

УТВЕРЖДАЮ

директор АНПОО «Образовательные технологии
«Скилбокс (Коробка навыков)»

Д. Р. Халилов

РАБОЧАЯ ПРОГРАММА ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

по профессиональному модулю

ПМ.08 Разработка дизайна веб-приложений

образовательной программы «Веб-разработка на Python» среднего
профессионального образования – программы подготовки
специалистов среднего звена по специальности 09.02.07
Информационные системы и программирование

Квалификация: разработчик веб и мультимедийных приложений

Форма обучения: очная

СОДЕРЖАНИЕ

1. ПАСПОРТ ПРОГРАММЫ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ
2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ
3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ
4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

1. ПАСПОРТ ПРОГРАММЫ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

1.1. Область применения программы производственной практики

Программа производственной практики профессионального модуля ПМ.08 Разработка дизайна веб-приложений является составной частью программы подготовки специалистов среднего звена по специальности СПО в соответствии с ФГОС по специальности 09.02.07 Информационные системы и программирование в части освоения основного вида профессиональной деятельности “Разработка дизайна веб-приложений”.

1.2. Цели и задачи программы производственной практики — требования к результатам освоения программы производственной практики

В результате прохождения производственной практики студент должен освоить основной вид деятельности “Разработка дизайна веб-приложений” и соответствующие ему профессиональные компетенции.

1.2.1 Перечень профессиональных компетенций:

ПК-8.1. Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика.

ПК-8.2. Формировать требования к дизайну веб-приложений на основе анализа предметной области и целевой аудитории.

ПК-8.3. Осуществлять разработку дизайна веб-приложения с учетом современных тенденций в области веб-разработки.

1.2.2. Результаты прохождения производственной практики

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями обучающийся в ходе освоения программы производственной практики должен:

1. В соответствии с минимальными требованиями к результатам освоения вида деятельности “Разработка дизайна веб-приложений” по ФГОС:

а. **Приобрести практический опыт** в разработке дизайна веб-приложений в соответствии со стандартами и требованиями заказчика; создании, использовании и оптимизировании изображений для веб-приложений; разработке интерфейса пользователя для веб-приложений с использованием современных стандартов.

б. **Уметь** создавать, использовать и оптимизировать изображения для веб-приложений; выбирать наиболее подходящее для целевого рынка дизайнерское решение; создавать дизайн с применением промежуточных эскизов, требований к эргономике и технической эстетике; разрабатывать интерфейс пользователя для веб-приложений с использованием современных стандартов.

с. **Знать** нормы и правила выбора стилистических решений; современные методики разработки графического интерфейса; требования и нормы подготовки и использования изображений в информационно-телекоммуникационной сети "Интернет" (далее - сеть Интернет); государственные стандарты и требования к разработке дизайна веб-приложений.

2. **В соответствии с Перечнем знаний, умений, навыков в соответствии со спецификацией стандарта демонстрационного экзамена КОД 1.4 раздел 3 "Графический дизайн веб-страниц", специалист должен знать и понимать:**

структуру и общепринятые элементы веб-страниц различных видов и назначений; вопросы, связанные с когнитивными, социальными, культурными, технологическими и экономическими условиями при разработке дизайна; основные принципы создания дизайна по предоставляемым инструкциям и спецификациям; основные принципы организации контента веб-приложения; основные правила выбора цвета, работы с типографикой и композицией; принципы и методы создания и адаптации графики для использования ее на веб-сайтах; правила поддержания фирменного стиля, бренда и стиливых инструкций; ограничения, которые накладывают мобильные устройства и разрешения экранов при использовании их для просмотра веб-сайтов; принципы построения эстетичного и креативного дизайна; современные стили и тенденции дизайна.

1.3. Количество часов на освоение программы производственной практики

Программа производственной практики ПМ.08 Разработка дизайна веб-приложений рассчитана на прохождение студентами практики в объеме 100 часов.

Распределение разделов и тем по часам приведено в тематическом плане.

2. СТРУКТУРА И СОДЕРЖАНИЕ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

Практическая подготовка осуществляется в «ОТ «Скилбокс» на образовательной платформе <https://go.skillbox.ru/> исключительно с применением дистанционных образовательных технологий и электронного обучения.

2.1. Объем производственной практики и виды производственной работы

Виды работ	Объем часов	из них в форме практической подготовки
Производственная практика (всего)	125	125
в том числе:		
● ознакомительная деятельность	8	8
● практическая деятельность	109	109
● оформление отчетной документации	8	8
Промежуточная аттестация в форме дифференцированного зачета		

2.2. Тематический план и содержание производственной практики

Наименование разделов, виды деятельности	Содержание производственной практики	Кол-во часов
Раздел 1. Ознакомительная часть практики	Цель и задачи практики. Выдача заданий на практику. Характеристика и общие сведения о проекте.	4
	Инструктаж обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка.	4
Раздел 2. Практическая деятельность	<p>Виды работ.</p> <ol style="list-style-type: none"> 1. Ознакомление с комплексным заданием в формате кейса (Образец задания приведен в Приложении 1) 2. Выделение в комплексном задании видов деятельности и задач входящих в текущую производственную практику (Разработка дизайна веб-приложений) 3. Выполнение выделенных из комплексного задания задач. 4. Защита результатов выполнения задач. 	109
Раздел 3. Оформление отчетной	Обобщение материалов практики, подготовка отчета по практике (в т.ч. Дневник)	6
	Сдача отчета по практике. Дифференцированный зачет.	2

документ ации по практике		
		125

В соответствии с положением о практической подготовке обучающиеся:

- направляются на практику в соответствии с распорядительным актом образовательной организации;
- при направлении на практику обучающиеся получают установочные данные: цели и задачи практики, а также индивидуальное или групповое задание по освоению компетенций, соответствующих виду деятельности “Разработка дизайна веб-приложений”;
- проходят инструктаж обучающихся по ознакомлению с требованиями охраны труда, техники безопасности, пожарной безопасности, а также правилами внутреннего трудового распорядка;
- в период практики выполняют полученное задание и оформляют отчет (форму отчета и конкретное задание обучающиеся получают при направлении на практику);
- по окончании практики обучающиеся защищают отчет о прохождении практики преподавателю.

3. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

3.1. Требования к минимальному материально-техническому обеспечению

Производственная практика проводится в АНПОО «Образовательные технологии «Скилбокс» (Коробка навыков)», а также в организациях по профилю специальности 09.02.07 Информационные системы и программирование на основе договоров, включая сетевые, заключаемых между «ОТ «Скилбокс» и этими организациями.

Практическая подготовка реализуется на основании договора о практической подготовке обучающихся № 14062023 от 14 июня 2023 года с Обществом с ограниченной ответственностью «ВК».

Производственную практику можно проводить как концентрированно, так и рассредоточено.

Оборудование организаций/предприятий должно включать индивидуальные персональные компьютеры.

1. Операционная система Windows, Linux, MacOS официально поддерживаемая разработчиком на момент прохождения практики
2. Браузер Google Chrome, версии последней на момент прохождения практики
3. Учетная запись Google
4. Офисный пакет Google Workspace

3.2. Информационное обеспечение обучения, организации и проведения производственной практики

3.2.1 Нормативные акты:

- Федеральный закон от 29.12.2012 № 273-ФЗ «Об образовании в Российской Федерации»

- Приказ Министерства науки и высшего образования Российской Федерации № 885, Министерства просвещения Российской Федерации № от 390 05.08.2020 "О практической подготовке обучающихся". (Зарегистрирован в Минюсте Российской Федерации 11.09.2020 № 59778)

- федеральный государственный образовательный стандарт среднего профессионального образования по профессии 09.02.07 Информационные системы и программирование.

- Положение о формах, периодичности и порядке проведения текущего контроля успеваемости и промежуточной аттестации обучающихся «ОТ «Скилбокс».

3.2.2 Перечень рекомендуемых учебных изданий, Интернет-ресурсов, дополнительной литературы.

3.2.2.1. Основные электронные издания (электронные ресурсы)

1. Немцова, Т. И. Компьютерная графика и web-дизайн : учебное пособие / Т.И. Немцова, Т.В. Казанкова, А.В. Шнякин ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 400 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0790-0. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1905248> (дата обращения: 07.04.2023). – Режим доступа: по подписке.
2. Немцова, Т. И. Практикум по информатике. Компьютерная графика и web-дизайн : учебное пособие / Т.И. Немцова, Ю.В. Назарова ; под ред. Л.Г. Гагариной. — Москва : ФОРУМ : ИНФРА-М, 2023. — 288 с. + Доп. материалы [Электронный ресурс]. — (Среднее профессиональное образование). - ISBN 978-5-8199-0800-6. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1908342> (дата обращения: 07.04.2023). – Режим доступа: по подписке.

3.2.2.2. Дополнительные источники

1. Дадьянова И.Б. Современные технологии анимации в веб-дизайне // Культура и искусство. 2022. №2. URL: <https://cyberleninka.ru/article/n/sovremennye-tehnologii-animatsii-v-veb-dizayne> (дата обращения: 07.04.2023).
2. Исмойилов Х.Б. Стили сайтов в веб-дизайне // Современные материалы, техника и технологии. 2018. №2 (17). URL: <https://cyberleninka.ru/article/n/stili-saytov-v-veb-dizayne> (дата обращения: 07.04.2023).
3. Калугина Ю.В., Кондакова А.А., Михайлов А.С., Стрельникова С.В. Роль цвета в веб-дизайне // Решетневские чтения. 2018. №. URL: <https://cyberleninka.ru/article/n/rol-tsveta-v-veb-dizayne> (дата обращения: 07.04.2023).

4. Китаевская Т.Ю. Альтернативные стили в веб-дизайне // Вестник российских университетов. Математика. 2014. №2. URL: <https://cyberleninka.ru/article/n/alternativnyye-stili-v-veb-dizayne> (дата обращения: 07.04.2023).
5. Морозов М.Д., Романов В.В. Роль эстетики в веб-дизайне // Вестник АГТУ. 2019. №2 (68). URL: <https://cyberleninka.ru/article/n/rol-estetiki-v-veb-dizayne> (дата обращения: 07.04.2023).

4. КОНТРОЛЬ И ОЦЕНКА РЕЗУЛЬТАТОВ ОСВОЕНИЯ ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ

Контроль и оценка результатов освоения производственной практики осуществляется руководителем практики от предприятия и руководителем практики от образовательной организации.

Код и наименование профессиональных и общих компетенций, формируемых в рамках производственной практики	Основные показатели оценки результата	Формы и методы контроля и оценки
ПК-8.1. Разрабатывать дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика.	Оценка “отлично”: <ul style="list-style-type: none">● Разработаны дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика.● Через защиту результатов и отчет о практике продемонстрирована полная сформированность требуемых знаний, умений, практического опыта:● Цели, задачи и результаты производственной практики полностью выполнены. Оценка “хорошо”: <ul style="list-style-type: none">● Разработаны дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика.● Через защиту результатов и отчет о практике	Защита результатов выполнения задания; отчет о практике; интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики.

	<p>продемонстрировано сформированность большей части требуемых знаний, умений, практического опыта.</p> <ul style="list-style-type: none"> ● Цели, задачи и результаты производственной практики в целом или с незначительными ошибками выполнены. <p>Оценка “удовлетворительно”:</p> <ul style="list-style-type: none"> ● Разработаны дизайн-концепции веб-приложений в соответствии с корпоративным стилем заказчика. ● Через защиту результатов и отчет о практике продемонстрировано частичная сформированность требуемых знаний, умений, практического опыта. ● Цели, задачи и результаты производственной практики частично или со значительными ошибками выполнены. 	
<p>ПК-8.2. Формировать требования к дизайну веб-приложений на основе анализа предметной области и целевой аудитории.</p>	<p>Оценка “отлично”:</p> <ul style="list-style-type: none"> ● Сформированы требования к дизайну веб-приложений на основе анализа предметной области и целевой аудитории. ● Через защиту результатов и отчет о практике продемонстрирована полная сформированность требуемых знаний, умений, практического опыта: ● Цели, задачи и результаты учебной практики полностью выполнены. <p>Оценка “хорошо”:</p> <ul style="list-style-type: none"> ● Сформированы требования к дизайну веб-приложений на основе анализа предметной области и целевой 	<p>Защита результатов выполнения задания; отчет о практике; интерпретация результатов наблюдений за деятельностью обучающегося в процессе практики.</p>

	<p>аудитории.</p> <ul style="list-style-type: none"> ● Через защиту результатов и отчет о практике продемонстрировано сформированность большей части требуемых знаний, умений, практического опыта. ● Цели, задачи и результаты учебной практики в целом или с незначительными ошибками выполнены. <p>Оценка “удовлетворительно”:</p> <ul style="list-style-type: none"> ● Сформированы требования к дизайну веб-приложений на основе анализа предметной области и целевой аудитории. ● Через защиту результатов и отчет о практике продемонстрировано частичная сформированность требуемых знаний, умений, практического опыта. ● Цели, задачи и результаты учебной практики частично или со значительными ошибками выполнены. 	
<p>ПК-8.3. Осуществлять разработку дизайна веб-приложения с учетом современных тенденций в области веб-разработки.</p>	<p>Оценка “отлично”:</p> <ul style="list-style-type: none"> ● Осуществлена разработку дизайна веб-приложения с учетом современных тенденций в области веб-разработки. ● Через защиту результатов и отчет о практике продемонстрирована полная сформированность требуемых знаний, умений, практического опыта: ● Цели, задачи и результаты производственной практики полностью выполнены. <p>Оценка “хорошо”:</p>	<p>Защита результатов выполнения задания; отчет о практике; интерпретация результатов наблюдений за деятельностью обучающегося в</p>

	<ul style="list-style-type: none">● Осуществлена разработку дизайна веб-приложения с учетом современных тенденций в области веб-разработки.● Через защиту результатов и отчет о практике продемонстрировано сформированность большей части требуемых знаний, умений, практического опыта.● Цели, задачи и результаты производственной практики в целом или с незначительными ошибками выполнены. <p>Оценка “удовлетворительно”:</p> <ul style="list-style-type: none">● Осуществлена разработку дизайна веб-приложения с учетом современных тенденций в области веб-разработки.● Через защиту результатов и отчет о практике продемонстрировано частичная сформированность требуемых знаний, умений, практического опыта.● Цели, задачи и результаты производственной практики частично или со значительными ошибками выполнены.	процессе практики.
--	---	--------------------

Образец задания

Над чем предстоит работать

Вы реализуете весь бэкенд клона Twitter: поработаете с дизайном API, напишете бэкенд на одном из изученных в рамках курса фреймворков — FastAPI или Flask. Поработаете с базой данных Postgres, покроете приложение unit-тестами и развернёте приложение через Docker Compose.

Какие материалы можно использовать

Основная информация дана в техническом задании проекта, которое прикреплено ниже. Ознакомьтесь с ним и приступите к выполнению. Также в прикреплённых материалах есть вёрстка для фронтенда, которую нужно прикрутить к своему коду.

Тема: «Сервис микроблогов».

Легенда

Вы — главный бэкенд-разработчик на Python известной веб-студии и уже привыкли к тому, что полностью отвечаете за бэкенд. Архитектура приложения, реализация бизнес-логики, тестирование и развёртывание — всё то, что вы любите и умеете отлично делать. Основная специализация веб-студии — создание лендингов, айдентики, CMS, поэтому вы легко можете в одиночку вести несколько проектов, так как основная нагрузка ложится на фронтенд. В итоге ваша работа выглядит так: получить описание задачи от CEO, договориться или получить уже готовый контракт API от фронтенд-разработчика и реализовать бэкенд под этот фронт.

И вот на волне импортозамещения один местный бизнесмен, который слышал о качестве работы вашей студии, решил прийти к вам с заказом на реализацию собственного корпоративного сервиса микроблогов, очень похожего на Twitter. Таких заказов сейчас невероятно много, поэтому вы оказались загружены, а согласование требований и контракта прошло заочно: вам предоставили техническое задание. Поскольку вы были заняты на другом проекте, а фронтенд-разработчик был свободен, вы решили поручить ему разработку контракта, что он успешно сделал, успев написать ещё и фронтенд.

В итоге, когда вы приступили к этому проекту, в вашем распоряжении были готовый фронтенд, описание API и горящие сроки (как же без этого). Теперь ваша очередь показать класс и написать бэкенд под этот непростой проект.

Техническое задание

Цель задачи

Реализовать бэкенд сервиса микроблогов.

Описание поведения

Для корпоративного сервиса микроблогов необходимо реализовать бэкенд приложения. Поскольку это корпоративная сеть, то функционал будет урезан относительно оригинала. Как правило, описание сервиса лучше всего дать через функциональные требования, то есть заказчик формулирует простым

языком, что система должна уметь делать. Или что пользователь хочет делать с системой. И вот что должен уметь делать наш сервис:

Функциональные требования:

1. Пользователь может добавить новый твит.
2. Пользователь может удалить свой твит.
3. Пользователь может зафолловить другого пользователя.
4. Пользователь может отписаться от другого пользователя.
5. Пользователь может отмечать твит как понравившийся.
6. Пользователь может убрать отметку «Нравится».
7. Пользователь может получить ленту из твитов отсортированных в порядке убывания по популярности от пользователей, которых он фоловит.
8. Твит может содержать картинку.

Заметим, что требования регистрации пользователя нет: это корпоративная сеть и пользователи будут создаваться не нами. Но нам нужно уметь отличать одного пользователя от другого. Об этом поговорим чуть позже.

Также систему описывают через список нефункциональных требований, то есть список того, как система должна выполнять функциональные требования.

Нефункциональные требования:

1. Систему должно быть просто развернуть через Docker Compose.

2. Система не должна терять данные пользователя между запусками.
3. Все ответы сервиса должны быть задокументированы через Swagger. Документация должна быть доступна в момент запуска приложения. Также не забудьте оформить развёрнутый README с описанием проекта и инструкцией по запуску приложения.

Состав продукта:

Основным потребителем нашего контента будет вот такая страничка:

Для этого нужно будет написать несколько endpoint.

Внимание: Поскольку мы разрабатываем сервис в рамках корпоративной системы, нам не придется самим реализовывать аутентификацию. По задумке пользователь будет переходить на наш сайт уже имея у себя ключ для авторизации. Все это будет делать браузер, скрытно от пользователя. Все endpoint ниже имеют http-header с названием api-key. Вот сюда frontend будет подставлять его. А пока, для демонстрации заказчику, фронтенд-разработчик сделал небольшую форму, в которую мы можем подставить такой ключ для пользователя на бэкенде. Так что мы сможем протестировать наше приложение, и показать клиенту как оно работает! Ключ вы можете подставить совершенно любой, главное чтобы по нему нашелся пользователь в вашей базе данных. Ну а какими записями вы наполните таблицы, чтобы продемонстрировать/протестировать приложение - решайте сами. Иначе говоря ваша задача, в том числе положить что-то в базу, чтобы показать работу приложения.

Первый и самый важный:

1.

POST /api/tweets

HTTP-Params:

api-key: str

{

“tweet_data”: string

“tweet_media_ids”: Array[int] // Опциональный параметр. Загрузка картинок будет происходить по endpoint /api/media. Фронтенд будет подгружать картинки туда автоматически при отправке твита и подставлять id оттуда в json.

}

Запросом на этот endpoint пользователь будет создавать новый твит.

Бэкенд будет его валидировать и сохранять в базу.

В ответ должен вернуться id созданного твита.

{

“result”: true,

“tweet_id”: int

}

2.

Endpoint для загрузки файлов из твита. Загрузка происходит через отправку формы.

POST /api/medias

HTTP-Params:

api-key: str

form: file="image.jpg"

В ответ должен вернуться id загруженного файла.

```
{  
  "result": true,  
  "media_id": int  
}
```

3.

Ещё нам потребуется endpoint по удалению твита. В этом endpoint мы должны убедиться, что пользователь удаляет именно свой собственный твит.

DELETE /api/tweets/<id>

HTTP-Params:

api-key: str

В ответ должно вернуться сообщение о статусе операции.

```
{  
  "result": true  
}
```

Пользователь может поставить отметку «Нравится» на твит.

POST /api/tweets/<id>/likes

HTTP-Params:

api-key: str

В ответ должно вернуться сообщение о статусе операции.

```
{  
  "result": true  
}
```

4.

Пользователь может убрать отметку «Нравится» с твита.

DELETE /api/tweets/<id>/likes

HTTP-Params:

api-key: str

В ответ должно вернуться сообщение о статусе операции.

```
{  
  "result": true  
}
```

5.

Пользователь может зафолловить другого пользователя.

POST /api/users/<id>/follow

HTTP-Params:

api-key: str

В ответ должно вернуться сообщение о статусе операции.

```
{
```

```
“result”: true  
}
```

6.

Пользователь может убрать подписку на другого пользователя.

DELETE /api/users/<id>/follow

HTTP-Params:

api-key: str

В ответ должно вернуться сообщение о статусе операции.

```
{  
“result”: true  
}
```

7.

Пользователь может получить ленту с твитами.

GET /api/tweets

HTTP-Params:

api-key: str

В ответ должен вернуться json со списком твитов для ленты этого пользователя.

```
{  
“result”: true,  
“tweets”: [  

```

```
{  
  "id": int,  
  "content": string,  
  "attachments" [  
    link_1, // relative?  
    link_2,  
    ...  
  ]  
  "author": {  
    "id": int  
    "name": string  
  }  
  "likes": [  
    {  
      "user_id": int,  
      "name": string  
    }  
  ]  
},  
...  
]
```

В случае любой ошибки на стороне бэкенда возвращайте сообщение

следующего формата:

```
{  
  "result": false,  
  "error_type": str,  
  "error_message": str  
}
```

8.

Пользователь может получить информацию о своём профиле:

GET /api/users/me

HTTP-Params:

api-key: str

В ответ получаем:

```
{  
  "result": "true",  
  "user": {  
    "id": "int",  
    "name": "str",  
    "followers": [  
      {  
        "id": "int",  
        "name": "str"  
      }  
    ]  
  }  
}
```

```
],  
"following":[  
  {  
    "id":"int",  
    "name":"str"  
  }  
]  
}  
}
```

9.

Пользователь может получить информацию о произвольном профиле по его

id:

GET /api/users/<id>

В ответ получаем:

```
{  
  "result":"true",  
  "user":{  
    "id":"int",  
    "name":"str",  
    "followers":[  
      {  
        "id":"int",
```

```
"name":"str"  
}  
],  
"following":[  
{  
"id":"int",  
"name":"str"  
}  
]  
}  
}
```

Технические требования

База данных: используйте базу данных PostgreSQL. Храните в ней информацию о твитах, пользователях, лайках, фоловерах и медиа.

Документация: как вы заметили, в ТЗ документация не совсем в обычном виде. Всё дело в том, что фронтенд-разработчик — новичок и не знает, как принято оформлять документы в вашей компании. Давайте это исправим. Все ответы сервиса должны быть задокументированы через Swagger.

Документация должна быть доступна в момент запуска приложения. Также не забудьте оформить развёрнутый README с описанием проекта и инструкцией по запуску приложения.

Деплой: запуск приложения должен происходить по команде `docker-compose up -d`.

Тестирование: приложение должно быть покрыто unit-тестами, проверено линтером.

Полуопционально: туру. Написать код так, чтобы он удовлетворил всем требованиям туру, сложно. Но постарайтесь, чтобы статический типизатор не слишком сильно ругался на ваш код. В идеале чтобы ошибок не было вообще.

Формат сдачи материалов и оценивание

Результат проведённой работы опубликуйте на GitLab и отправьте ссылку через форму ниже. Убедитесь, что предоставили доступ к репозиторию и проект полностью готов к тому, чтобы быть развёрнутым.

Критерии оценивания

- Проект полностью соответствует ТЗ. Все функциональные и нефункциональные требования имплементированы и протестированы. Приложение работает без ошибок (нет неотловленных exception).
- Проект легко развернуть в одну-две команды на любой машине.
- Pythonic-код легко читаем, идиоматичен, проверен линтерами и соответствует их требованиям. Есть вся необходимая документация функций, методов, классов и модулей, присутствует аннотация типов.
- Есть файл README.md, в котором для пользователя описывается подробная инструкция по эксплуатации сервиса.

Подсказки и советы

- Можете проверить качество тестов через метрику покрытия: сколько процентов кода покрыто тестами? Используйте для этого утилиту.
- Чтобы сделать красивый README.md, можете воспользоваться гайдлайном.
- Не забудьте разделить dev- и prod-окружение. Это значит, что в запущенном приложении, скорее всего, не понадобятся библиотеки линтеров и тестов.
- В документации Flask вы найдёте информацию о том, как делать загрузку файлов во flask-приложении. Во всех остальных фреймворках похожий принцип работы.
- В шаблон проекта уже внесены минимальные настройки для запуска UI сайта. Используйте их и сделайте другие по своему усмотрению.