

Проект 2 по модулю 3

С пятой по восьмую неделю модуля 3 вашим самостоятельным практическим заданием будет разработка REST-сервиса с использованием фреймворка FastAPI.

Предметная область

Прежде чем приступить к выполнению проекта, выберите предметную область. Это может быть:

- управление задачами (см. проект-референс из лекций);
- управление каталогом книг (см. Задание на программирование 5.7);
- любая другая предметная область, при условии что она подразумевает разработку реляционной БД и требует алгоритмического решения конкретной бизнес-задачи. Приветствуются оригинальные тематики, связанные с вашими научными интересами, хобби или коммерческими идеями.

! Важно

Переиспользование кода из лекций и практических заданий приветствуется, однако заимствование решений у других студентов строго запрещено. Ваш проект должен представлять собой уникальную работу, выполненную самостоятельно.

Содержание проекта

В проекте обязательно должны присутствовать следующие компоненты:

1. CRUD-операции

- База данных.** Используйте любую реляционную СУБД (выбор СУБД не влияет на оценку, при условии что обеспечена ее работоспособность):
 - Если используете SQLite, убедитесь, что при запуске ваше приложение автоматически создает файл БД, отвечающий требованиям приложения.
 - Если используете PostgreSQL, MySQL или др. клиент-серверную СУБД, воспользуйтесь Docker Compose для создания контейнера приложения с необходимым образом СУБД. Подробности см. в лекции 4-й недели.
 - Создайте структуру БД, включающую не менее трех связанных таблиц.
 - Применение миграций для управления изменениями схемы БД будет дополнительным преимуществом (однако не влияет на оценку).
- REST API.** Реализуйте полный набор REST-эндпоинтов для создания, чтения, обновления и удаления записей в БД с учетом связей между сущностями. Для реализации REST API **обязательно** использовать фреймворк FastAPI. Выбор вспомогательных библиотек не регламентирован.

2. **Аутентификация и авторизация.** Реализуйте эндпоинты для регистрации новых пользователей и для входа пользователей в систему. Реализуйте как минимум один эндпоинт, который требует авторизации через токен доступа.
3. **Решение бизнес-задачи.** Разработайте эндпоинт, который решает конкретную задачу в рамках выбранной предметной области, используя алгоритмический компонент (например, расчет рейтинга, анализ данных, оптимизация процессов). Например, в контексте управления задачами вы можете разработать функцию, предлагающую оптимальный способ назначения задач сотрудникам с учетом их загруженности и уровня квалификации (можно применить методы динамического программирования или другие алгоритмы). Решение должно принимать на вход параметры через HTTP-запрос и выдавать ответ в виде JSON. Если решение представляет собой простую CRUD-операцию без расширенной логики, баллы за этот раздел получить нельзя.
4. **Тесты.** Снабдите сервис автоматическими тестами на основе TestClient. Разработка тестовых сценариев для Postman или аналогичного ПО будет дополнительным преимуществом.

Сдача проекта и критерии оценивания

В качестве сданного проекта вам нужно предоставить следующие материалы:

1. Ссылку на демо-запись с презентацией вашего проекта (5–7 минут).
2. Отчет утилиты Pylint о качестве кода. Выполните команду `pylint app > pylint.txt` и приложите файл `pylint.txt` к заданию в LMS
3. Ссылку на репозиторий GitHub с кодом вашего проекта. Убедитесь, что репозиторий содержит файл `README.md` с подробными инструкциями по запуску приложения, а также необходимые файлы с информацией о зависимостях (`requirements.txt`, `docker-compose.yml` в случае использования Docker и т. п.)

Разворачивать сервис в Интернете необязательно; для демонстрации его работы вы можете запускать его локально.

Проект оценивается по следующим параметрам:

1. Корректность работы всех API-методов.
2. Качество кода и использование лучших практик FastAPI.
3. Реализация всех функциональных требований.

Подробные критерии оценивания итогового проекта доступны [по ссылке](#).

Запись экрана в Zoom для итогового проекта

Подготовка

Для записи демо вам потребуются:

- установленный и обновлённый до последней версии Zoom,
- разрешение записи экрана в Zoom,
- разрешение записи звука в Zoom,
- разрешение записи картинки с камеры в Zoom.

Запись

1. Начните конференцию Zoom на одного человека (где будете только вы! иначе Zoom выставит вам ограничение в 40 минут на конференцию).
2. Включите камеру и микрофон. Ни то, ни другое **нельзя** выключать до конца записи.
3. Включите демонстрацию экрана. В открывшемся выборе окна выберите **весь экран (desktop)**. Менять демонстрацию до конца записи **тоже нельзя**.
4. Начните запись встречи. Неважно, куда вы будете записывать — на облако или в локальное хранилище (тем не менее, рекомендуем записывать локально, чтобы запись была на вашем компьютере; если у вас платный аккаунт Zoom, который предусматривает облачное хранение, можно писать и на облако).
5. Ваш сервис не обязательно разворачивать в Интернете, можно запустить его локально.
6. Проведите демонстрацию работы вашего сервиса. Если вы выбрали дизайн систему, расскажите, какую, и покажите её сайт. Расскажите, что за сервис вы сделали, как он работает, какие функции задействуются во время работы с той или иной страницей.
7. Во время выполнения работы **можно** открывать веб-браузер и редактор кода, **нельзя** открывать какие бы то ни было соцсети (считаются даже обыкновенные оповещения, поэтому закройте всё заранее), Google и любые другие сайты. Единственное исключение — показать аналогичный функционал в другом сервисе.
8. Помните про ограничение в 5–7 минут! **За несоблюдение тайминга оценка будет снижена.**

Окончание записи

9. Когда вы закончите демонстрацию, выключите запись и выйдите из конференции.

10. Полученную запись загрузите на любое облако (Google Drive, Яндекс.Диск, Dropbox и т. д.).
11. Запись сделайте доступной по ссылке.
12. Ссылку **из п.11** предоставьте в ответе при сдаче проекта в LMS.