ПРИЛОЖЕНИЕ 2

ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Департамент программной инженерии

СОГЛАСОВАНО
Научный руководитель,
доцент департамента
программной инженерии
факультета компрьютерных наук,
канд. физико-математических наук
_____ Г.Н. Жукова
« _____ » _____2023 г.

УТВЕРЖДАЮ
Академический руководитель
образовательной программы
«Программная инженерия»
профессор департамента программной
инженерии, канд. техн. наук
_____ В.В. Шилов
« _____ » _____2023 г.

СЕРВЕРНАЯ ЧАСТЬ ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ И УПРАВЛЕНИЯ ИНДИВИДУАЛЬНЫМИ СТУДЕНЧЕСКИМИ ЗАДАНИЯМИ ДЛЯ КУРСА «КОМПЬЮТЕРНЫЙ ПРАКТИКУМ ПО АЛГЕБРЕ НА PYTHON»

Руководство программиста

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.13-62 РП 01-1

Исполнитель :
Студент группы БПИ196
_____/ Сахаров Никита
«_____»_____2023 г.

2023

УТВЕРЖДЁН
RU.17701729.04.13-62 РП 01-1-ЛУ

СЕРВЕРНАЯ ЧАСТЬ ПРИЛОЖЕНИЯ ДЛЯ ГЕНЕРАЦИИ И УПРАВЛЕНИЯ ИНДИВИДУАЛЬНЫМИ СТУДЕНЧЕСКИМИ ЗАДАНИЯМИ ДЛЯ КУРСА «КОМПЬЮТЕРНЫЙ ПРАКТИКУМ ПО АЛГЕБРЕ НА PYTHON»

Руководство программиста

RU.17701729.04.13-62 РП 01-1-ЛУ

Листов 12

2023

Содержание

1 Назначение программы

Серверная часть веб-приложения предназначена для обработки запросов от клиентской части, управления базой данных, выполнения бизнес-логики приложения, а также обеспечения безопасности и управления пользователями.

Основными функциями сервера являются:

- Обработка запросов на просмотр, создание, редактирование и удаление данных, связанных с лабораторными работами.
- Управление учетными записями пользователей, включая аутентификацию и авторизацию.
- Обработка загрузки и скачивания файлов, связанных с лабораторными работами.
- Выполнение бизнес-логики приложения, включая генерацию уникальных вариантов заданий.

Таким образом, серверная часть приложения является ключевым элементом в обеспечении функциональности веб-приложения для автоматизации проведения лабораторных работ.

2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

2.1 Требования к составу и параметрам технических средств

- Операционная система: Ubuntu 18.04 или более поздняя версия, а так же операционные системы поддерживающие менеджер пакетов pipenv.
- Оперативная память: Зависит от числа rps. Рекомендуемо 1ГБ оперативной памяти.
- Процессор с тактовой частотой не ниже 2.0ГГц. Рекомендуется использовать как минимум двухядерный процессор.
- Размер дискового пространства зависит от количества пользователей о которых предполагается хранить данные.

2.2 Требования к информационной и программной совместимости

Операционная система Ubuntu 18.04 или любая другая поддерживающая менеджер пакетов pipenv, с устаноленными пакетами:

- **fastapi**
- **uvicorn**
- **bcrypt**
- **sqlalchemy**
- **asyncpg**
- **httpx**
- **fastapi-security**
- **pydantic**
- **databases**
- **python-dotenv**
- **python-multipart**
- **gevent**
- **psutil**
- **greenlet**
- **psycopg2-binary**
- **pyjwt**

- **autopep8**
- **fastapi-cache**
- **pytest**
- **testsuite**
- **pytest-asyncio**
- **aiosqlite**
- **flake8**
- **jupyterlab**
- **pytz**
- **black**
- **python-rocksdb**
- **numpy**
- **sympy**
- **locust**

3 Установка и запуск

3.1 Установка

Для установки и запуска сервера необходимо выполнить следующие шаги:

- Получите исходный код сервера, клонировав репозиторий с GitHub: https://github.com/Sugarhl/zxcursed_work.
- Установите необходимые зависимости, используя менеджер пакетов Pipenv. Для этого предоставлен скрипт `/scripts/install_deps.sh` в репозитории. Запустите его для автоматической установки зависимостей и активации виртуального окружения.
- В случае проблем с установкой Pipenv следуйте официальной документации для ручной установки [9].

3.2 Настройка базы данных

В проекте использовалась облачная база данных PostgreSQL, размещенная на Yandex.Cloud. Однако стоит заметить, что развертывание базы данных может быть организовано разными способами, например, с помощью Docker или других облачных провайдеров. В данном документе будет описан процесс настройки облачной базы данных на примере Yandex.Cloud:

- Зарегистрируйтесь на сайте Yandex.Cloud и создайте новый проект.
- В рамках этого проекта создайте новую базу данных PostgreSQL. В процессе создания базы данных настройте необходимые параметры, такие как версия PostgreSQL, объем и тип дискового пространства, конфигурация резервного копирования и т.д.
- Запустите базу данных и создайте пользователя с соответствующими правами для работы с созданной базой данных.
- Получите строку подключения к базе данных. Она будет выглядеть примерно так: `postgresql:/`
- Эту строку подключения следует внести в файлы `.env.test` и `.env.prod` в полях `DATABASE_URL` и `SYNC_DATABASE_URL`.

Если вы используете другого облачного провайдера, обратитесь к соответствующей документации.

3.3 Настройка секретов

После установки необходимо настроить файлы с секретами для сервера. В частности, необходимо создать и настроить два файла: `.env.test` и `.env.prod`. В них должны быть следующие параметры:

- `SECRET_KEY`: Секретный ключ для подписи JWT-токенов.

- `ALGORITHM`: Алгоритм, используемый для подписи JWT-токенов.
- `ACCESS_TOKEN_EXPIRE_MINUTES`: Время жизни доступного токена в минутах.
- `DATABASE_URL`: URL для асинхронного подключения к базе данных PostgreSQL.
- `SYNC_DATABASE_URL`: URL для синхронного подключения к базе данных PostgreSQL.

Убедитесь, что все URL-адреса баз данных и секретные ключи сохранены в безопасности и не разглашаются. Параметры критически важны для безопасности и правильной работы сервера.

3.4 Развертывание сервера

Сервер проекта разворачивается с использованием предложенного скрипта `deploy.sh`. Этот скрипт выполняет следующие операции:

- Удаляет существующий файл `.env`, если таковой имеется.
- Копирует содержимое файла `.env.prod` в новый файл `.env`. Это означает, что настройки сервера будут взяты из файла для продакшн-окружения.
- Запускает сервер приложения FastAPI с использованием сервера Uvicorn. Приложение будет доступно на порту 8000, а таймаут для keep-alive соединений составит 10 секунд.

Этот скрипт должен быть запущен в корневом каталоге проекта. После запуска сервера, приложение будет доступно для входящих подключений.

Обратите внимание, что скрипт подразумевает, что все необходимые зависимости уже установлены, а файл `.env.prod` содержит корректные настройки.

4 ОБРАЩЕНИЕ К ПРОГРАММЕ

4.1 Описание запросов

Для взаимодействия с сервером используется набор запросов, основанных на протоколе HTTP. Для более детального описания API используется OpenAPI спецификация, доступ к которой можно получить, отправив GET запрос на эндпоинт `/openapi.json` сервера.

Вы можете использовать инструмент командной строки, такой как `curl`, для отправки этого запроса. Общий формат для запроса схемы запросов выглядит следующим образом:

```
curl http://<hostname>:<port>/openapi.json
```

где `<hostname>` и `<port>` - это адрес и порт, на котором работает сервер.

В случае использования демонстрационного сервера, запрос для получения схемы будет выглядеть так:

```
curl http://158.160.4.55:8000/openapi.json
```

Обратите внимание, что адрес и порт сервера могут измениться в зависимости от конкретной конфигурации и окружения развертывания.

В полученном JSON-документе будут подробно описаны все доступные эндпоинты, включая описание возможных параметров запроса, ожидаемые ответы и ошибки. Эта информация предоставляет полное понимание работы с сервером приложения.

4.2 Запуск тестов

Тесты для приложения реализованы с использованием фреймворка pytest и расположены в директории ./tests. Тестирование включает в себя функциональное тестирование API, тестирование моделей данных и нагрузочное тестирование.

Для запуска тестов используется скрипт runtests.sh, расположенный в директории ./scripts. Этот скрипт автоматически обнаруживает и выполняет все тестовые случаи, находящиеся в директории с тестами. Кроме того, он выполняет настройку окружения для тестирования, включая установку переменных окружения из файла .env.test. Для запуска тестов выполните следуюшие действия:

- Откройте терминал.
- Перейдите в корневую директорию проекта.
- Запустите скрипт runtests.sh:

Перед запуском тестов убедитесь, что у вас установлены все необходимые зависимости и что файл .env.test содержит актуальные настройки.

Запуск тестов полностью автоматизирован и не требует дополнительных действий со стороны пользователя. Все результаты тестов выводятся в консоль, предоставляя подробную информацию о каждом тестовом случае. При успешном прохождении всех тестов выводится сообщение о том, что все тесты прошли без ошибок.

5 Поддержка и обслуживание

5.1 Слежение за кластером PostgreSQL в Яндекс Облако

Для обеспечения надлежащей работы кластера PostgreSQL в облачном сервисе Яндекс Облако рекомендуется использовать мониторинг состояния базы данных. Мониторинг позволяет получать информацию о производительности и доступности кластера, а также оперативно реагировать на возможные проблемы. Вот основные аспекты мониторинга состояния кластера:

1. **Метрики производительности:** С помощью мониторинга вы можете отслеживать такие показатели, как загрузка процессора, использование памяти, ввод-вывод и сетевая активность. Это поможет вам определить возможные узкие места и бутылочные горлышки в работе кластера.
2. **Логирование и ошибки:** Важно отслеживать журналы ошибок и предупреждений PostgreSQL для своевременного обнаружения и анализа возможных проблем. Мониторинг позволяет получать оповещения о появлении ошибок, и вы можете проводить анализ журналов для выявления причин возникших проблем.
3. **Производительность запросов:** С помощью мониторинга можно отслеживать долгие и медленные запросы, а также определять потенциальные узкие места в выполнении запросов. Это поможет вам оптимизировать работу с базой данных и повысить производительность.

Рекомендуется регулярно анализировать получаемую информацию и принимать соответствующие меры для оптимизации работы кластера PostgreSQL в Яндекс Облако.

5.2 Добавление нового генератора

Для добавления нового генератора в проект необходимо выполнить следующие шаги:

1. Определите новый тип генератора в перечислении `GenType`. Это можно сделать путем добавления нового элемента с указанием уникального значения для нового типа генератора.
2. Создайте новый класс генератора, наследующийся от базового класса `NotebookGenerator`. Реализуйте метод `generate_notebooks`, который будет генерировать нужное количество вариантов и возвращать их в виде списка объектов класса `Variant`. Внутри этого метода вы можете использовать соответствующую логику для генерации новых вариантов.
3. Для специальных задач генератора добавьте файл в модуль `tasks` с функциями генерации заданий. В этом файле можно реализовать функции, которые будут вызываться из метода `generate_notebooks` для создания специфических задач или компонентов вариантов.
4. В функции `get_generator_by_type` добавьте новую ветку `elif`, где будет проверяться переданный тип генератора, и если он соответствует новому типу, создайте экземпляр нового генератора и верните его.
5. В функции `generate_for_group` обновите логику выбора генератора. Добавьте новый тип генератора в условные операторы и создайте экземпляр нового генератора, если переданный тип соответствует новому типу генератора.

5.3 Рекомендации по поддержке MongoDB в проекте

Для обеспечения правильной поддержки MongoDB в проекте и замены предыдущего хранилища RocksDB, рекомендуется следовать следующим рекомендациям:

1. Установите MongoDB и настройте соединение с базой данных, указав необходимые параметры соединения, такие как адрес сервера MongoDB, порт и учетные данные.

2. Создайте новый класс хранилища, который наследуется от базового класса файлового хранилища (`FileStorage`). В этом классе реализуйте методы `save_file`, `get_file` и `delete_file`, чтобы они использовали функциональность работы с MongoDB.

3. Обновите соответствующие классы и методы в вашем проекте, чтобы они использовали новый класс хранилища, наследующийся от базового класса `FileStorage`. Замените вызовы методов работы с предыдущим хранилищем на соответствующие методы работы с MongoDB.

4. Учтите особенности и рекомендации работы с MongoDB при разработке новых функций и модулей в проекте. Включите соответствующие инструкции и обработку ошибок, связанных с взаимодействием с MongoDB, для обеспечения надежности и стабильности работы вашего приложения.

5. При необходимости, проведите тестирование и отладку ваших изменений, связанных с использованием MongoDB. Убедитесь, что все функции работают корректно и ожидаемым образом, и что данные правильно сохраняются, получаются и удаляются из базы данных MongoDB.

6 ИСТОЧНИКИ, ИСПОЛЬЗОВАННЫЕ ПРИ РАЗРАБОТКЕ

1. ГОСТ 19.103-77 Обозначения программ и программных документов. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.

2. ГОСТ 19.104-78 Основные надписи. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.

3. ГОСТ 19.105-78 Общие требования к программным документам. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.

4. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.

5. ГОСТ 19.201-78 Пояснительная записка. Требования к содержанию и оформлению. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.

6. ГОСТ 19.603-78 Общие правила внесения изменений. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.

7. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. //Единая система программной документации. –М.: ИПК Издательство стандартов, 2001.

8. Postgresql [Электронный ресурс]/ Хабр. [2018—2019].-Электрон. текстовые дан. -Режим доступа: https://habr.com/ru/post/340460/ (дата обращения: 18.04.2023).

9. Pipenv [Электронный ресурс]/ [2018—2019].-Электрон. текстовые дан. -Режим доступа: https://docs.pipenv.org/install/#installing-pipenv (дата обращения: 30.05.2023).

7 ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

Номера листов(страниц)					Всего листов (страниц в докум.)	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
Изм.	Измененных	Замененных	Новых	Архивированных					

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.13-62 РП				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата