**Тестовое задание**

Добрый день. Знаю, что выполнение тестовых заданий — это трудозатратный процесс, особенно когда у вас уже много заявок. Но если ничего не делать, то ничего и не получится. Отнеситесь к этому заданию просто как к лучшему пониманию и проверке себя. Успехов вам на вашем пути!

**Основная информация**

* Ваше ФИО: Досмухамед Абдыпатахов
* Сколько вам полных лет: 22
* Город проживания: Медина

**Задания**

1. Опишите свой самый интересный проект, связанный с разработкой на Python или Node.js. Если использовали при этом технологии искусственного интеллекта или машинного обучения, то обязательно это укажите. Постарайтесь детализировать:

* **Цель проекта**:

Один из самых интересных проектов, над которым я работал, был связан с разработкой системы управления профилями пользователей и задачами с использованием Node.js. В рамках этого проекта нужно было реализовать профили с такими характеристиками, как навыки, портфолио, платежные данные и задачи, а также систему категорий и подкатегорий для организации информации.

Цель проекта заключалась в том, чтобы создать гибкую систему управления пользователями, позволяющую им эффективно управлять своим профилем, добавлять задачи и отслеживать их прогресс. Мотивация проекта заключалась в создании структурированной базы данных и административной панели, через которую можно было бы легко управлять пользователями и категориями.

* **Ваш вклад**:

1. Профили пользователей:

- Я занимался разработкой функционала для работы с профилями, включая создание и обновление данных, таких как имя, фамилия, контактная информация и портфолио.

- Реализовал систему проверки и обновления данных, используя валидацию на серверной стороне для предотвращения ошибок и некорректных данных.

2. Управление категориями и подкатегориями:  
- Я создал систему категорий и подкатегорий для задач и проектов, используя ORM Sequelize для построения иерархических моделей.

- Создал контроллеры для управления категориями, добавив CRUD-функционал (создание, чтение, обновление и удаление), а также внедрил корректную обработку ошибок.

3. Система ролей и авторизация:

- Разработал систему ролей для администраторов, пользователей и других категорий с гибким распределением прав доступа.

- Внедрил JWT для аутентификации и авторизации пользователей, что обеспечило безопасный доступ к профилям и задачам на основании их ролей.

- Создал middleware для проверки ролей, чтобы гарантировать, что только пользователи с определёнными правами могут выполнять конкретные действия.

4. Логирование и мониторинг:

- Внедрил библиотеку winston для логирования всех основных действий пользователей, ошибок и важных событий в приложении.

- Логи позволяли отслеживать проблемы и проводить диагностику на основе пользовательских действий.

5. Пароли и безопасность:

- Использовал bcrypt для безопасного хэширования паролей и их хранения в базе данных.

- Для восстановления паролей была добавлена система, отправляющая пользователям ссылки для восстановления пароля на их email.

* **Технологии и инструменты**:

Node.js, Express.js, Sequelize, JWT, bcrypt, winston, PostgreSQL, Git, Docker

* **Сложности и их преодоление**: с какими трудностями столкнулись и как их решили.

- Одной из ключевых сложностей была реализация структуры категорий и подкатегорий, чтобы они могли быть взаимосвязаны. Для этого пришлось использовать рекурсивные запросы и правильно организовать связи в базе данных.

- В проекте была сложность с миграцией от использования import/export на require, что требовало значительных изменений в кодовой базе. Я адаптировал проект, переведя его на CommonJS, и это позволило устранить проблему с разными стандартами модулей.

- Еще одна задача — это грамотная реализация ролей с различными правами доступа. Для решения этой задачи я реализовал middleware, который проверял права доступа пользователей на основе их роли, что улучшило безопасность приложения.

* **Результаты и выводы**: чего достигли и чему научились в процессе работы.

- В результате удалось создать систему, которая эффективно управляет пользователями, профилями и задачами, а также обеспечивает гибкую настройку ролей и прав доступа.

- Я значительно улучшил свои навыки работы с ORM (Sequelize), JWT, и безопасностью на сервере.

2. Поделитесь, что сочтете нужным, о том, используете ли вы ИИ, о ваших целях, желаниях, идеях. Чувствуйте себя свободно.

- Я активно использую искусственный интеллект для обучения, повышения эффективности своей работы и решения сложных задач.

- Моя главная цель развиваться как профессионал в сфере разработки, постоянно повышая свои навыки в работе с технологиями, такими как Node.js.

3. Напишите простой чат-бот в Телеграм, который при присоединении говорит “Добрый день. Как вас зовут?”, берет ответ как имя и сообщает, например: “Рад знакомству, Алексей! Курс доллара сегодня 100р”, беря котировку из любого открытого источника. Используйте бесплатные решения, их достаточно. Опишите:

* Технологии, которые использовали.

NodeJs, axios, node-telegram-bot-api, nodemon.

* Встретившиеся проблемы и пути их решения.

Выбор пакета API для получения актуального курса валют.

* Инструкцию по подключению и запуску бота.

Первый шаг:

Откройте консоль и введите

npm install

Запустите проект:

npm start

<https://github.com/mukhammat/telegram-bot>