

Тестовое задание для кандидата на вакансию Junior Node.js Developer

Предлагаем вам продемонстрировать компании Сайберс ваши знания и выполнить тестовое задание (ТЗ).

Пожалуйста, обратите внимание на требования к технологиям, которые необходимо использовать, и качеству кода.

Если у вас возникнут вопросы по ТЗ, пишите нам или звоните по телефону +7 (383) 330-6626.

Выполненное ТЗ ждём на почту work@sibers.ru.

Желаем успехов!

ОБЯЗАТЕЛЬНЫЕ ТРЕБОВАНИЯ

Код выполненного задания должен быть структурирован, легко читаем и содержать необходимые комментарии. Все комментарии, инструкции и тексты должны быть написаны на английском (для демонстрации уровня владения языком). Обратите внимание на то, что реализация задания должна содержать все элементы законченной программы (обработка нестандартных ситуаций, некорректных действий пользователя и т. д.).

Постарайтесь достичь минимальной зависимости от версии Node.js (LTS-версия), Web-сервера, сервера базы данных и их настроек, и обеспечьте разделение логики и представления (шаблонизаторы или API от фронтенда).

Кроме того, для Web-интерфейсов:

- должна быть исключена возможность повреждения данных и самой базы любыми действиями пользователя, совершаемыми через Web-интерфейс (защита от SQL-инъекций, XSS, CSRF);
- необходимо обратить внимание на качество получаемого HTML кода (совместимость с распространенными современными браузерами, отделение представления от содержания).

ВРЕМЯ НА ВЫПОЛНЕНИЕ ТЕСТОВОГО ЗАДАНИЯ

В среднем выполнение подобной задачи у разработчика соответствующего уровня может занять до 8–16 часов.

Задача: Web-интерфейс для администрирования пользователей

Нужно создать Web-интерфейс для администрирования зарегистрированных пользователей сайта.

База данных содержит таблицу **users** с данными о зарегистрированных пользователях.

Набор полей таблицы:

1. Логин пользователя **username** (должен быть уникальным).
2. Пароль пользователя **password**.
3. Персональные данные — имя, фамилия, пол, дата рождения — **first_name**, **last_name**, **gender**, **birthdate**.

Интерфейс должен реализовывать следующую функциональность:

- просмотр списка зарегистрированных пользователей (с постраничной разбивкой и сортировкой по логину, имени, дате рождения);
- просмотр детальной информации по выбранному пользователю;
- добавление нового пользователя;
- редактирование существующей записи;
- удаление пользователя.

Интерфейс должен быть доступен пользователю с правами администратора только после прохождения аутентификации.

ТРЕБОВАНИЯ К ТЕХНОЛОГИЯМ

Предполагается, что для решения этой задачи кандидат будет использовать:

1. Backend:

- Node.js версии 16.x или выше;
- разрешены «лёгкие» фреймворки: Express.js, Кoa или Fastify;
- запрещены: NestJS, Angular Universal, Meteor и подобные.

2. СУБД:

- на выбор: PostgreSQL, MySQL, MongoDB или SQLite.

3. Frontend:

- чистый HTML/CSS/JavaScript или «лёгкие» библиотеки (React/Vue в однофайловых компонентах запрещены);
- допустимо: Handlebars, EJS, Pug для рендеринга на сервере.

4. Дополнительно:

- разрешены библиотеки: bcrypt, jsonwebtoken, Sequelize/Mongoose (ORM/ODM);
- запрещены готовые решения типа AdminJS.

ТРЕБОВАНИЯ К ФОРМАТУ ПРЕДСТАВЛЕНИЯ РЕЗУЛЬТАТА

Выполненное ТЗ должно быть собрано в архив в формате **.zip**. Допускается также использование сторонних ресурсов, не требующих наличия аккаунта для скачивания файлов. Например, github.com.

Результат выполнения ТЗ должен включать в себя:

1. SQL/NoSQL дампы базы данных со структурой и тестовыми данными (минимум 5 пользователей);
2. Исходный код (JavaScript, HTML, CSS);
3. Руководство по установке (README.md на английском):
 - установка зависимостей (**npm install**);
 - настройка подключения к базе данных (логин/пароль);
 - запуск приложения (**npm start**).

Критерии оценки

1. Работоспособность: весь функционал должен работать без ошибок.
2. Безопасность.
3. Качество кода:
 - чистый, модульный код;
 - использование промисов или **async/await** вместо колбэков.
4. Интерфейс:
 - удобная навигация, валидация форм;
 - адаптивность (корректное отображение на разных девайсах).