

# Тестовое задание для кандидата на вакансию C# (ASP.NET/Xamarin) Developer

Предлагаем вам продемонстрировать компании Сайберс ваши знания и решить приведённую ниже задачу. Дополнительные задания 1 и 2 расположены в порядке возрастания сложности и выполняются по желанию (при этом можно выполнить или оба или одно — любое из них).

Пожалуйста, обратите внимание на требования к технологиям, которые необходимо использовать, и качеству кода. Соблюдение всех требований не обязательно, но несоблюдение их будет минусом.

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

Выполненное тестовое задание ждём на почту [work@sibers.ru](mailto:work@sibers.ru). В теме письма не забудьте указать фамилию и имя, а также вакансию, на которую откликаетесь.

Желаем успехов!

## ОБЯЗАТЕЛЬНЫЕ ТРЕБОВАНИЯ

Код выполненного задания должен быть структурирован, легко читаем и содержать необходимые комментарии. Написанная вами программа должна являться законченным программным продуктом, т. е. должна легко устанавливаться, предусматривать обработку нестандартных ситуаций, быть устойчивой к некорректным действиям пользователей и т. д.

Задания можно выполнять:

- на Xamarin: .NET C# (.NET Standard (iOS или Android), SQLite);
- на Web: .NET C# (EntityFramework Code First, Web API, .NET Core).

## ТРЕБОВАНИЯ К ТЕХНОЛОГИЯМ

Нефункциональные требования:

- трёхуровневая архитектура проекта (уровень доступа к данным, уровень логики, уровень представления);
- в качестве хранилища данных желательно использовать MSSQL или SQLite;
- при использовании сторонних фреймворков и пакетов — они должны быть общедоступными, и желательно описать используемые версии в пояснительной записке;
- рекомендовано использовать units test для отладки уровня логики;
- для выполнения задания на ASP.NET желательно продемонстрировать использование клиентского скрипта на JavaScript;
- использовать как минимум версию .NET Core 5.0.

## ТРЕБОВАНИЯ К ФОРМАТУ ПРЕДСТАВЛЕНИЯ РЕЗУЛЬТАТА

Выполненное ТЗ должно быть собрано в архив и включать в себя:

1. Обязательно: с готовой программой предоставить все исходные файлы вместе с проектными файлами.
2. В случае необходимости — предоставить описание конфигурации системы, процесса запуска в пояснительной записке.

**Бинарные файлы отправлять не нужно.**

ВРЕМЯ НА ВЫПОЛНЕНИЕ ТЕСТОВОГО ЗАДАНИЯ

В среднем выполнение трёх подобных задач у разработчика соответствующего уровня может занять до 8–16 часов (на Xamarin) и 12–16 часов (на Web).

Задача: Программа для занесения данных о проекте в БД

Реализовать программу для занесения данных о проектах в базу данных и дальнейшей работы с этими данными.

Функциональные требования:

- возможность создавать/просматривать/редактировать/удалять информацию о проектах;
- возможность создавать/просматривать/редактировать/удалять информацию о сотрудниках;
- возможность добавлять и удалять сотрудников с проекта (один сотрудник может работать одновременно на нескольких проектах, и на одном проекте может работать несколько человек);
- для просмотра проектов предусмотреть различные способы фильтрации (по диапазону даты начала, по приоритетам, и т.п.) и сортировки (по основным полям).

Необходимая для хранения информация:

- название проекта,
- название компании-заказчика,
- название компании-исполнителя,
- данные о сотруднике (имя, фамилия, отчество, email человека),
- данные о руководителе проекта (один из сотрудников),
- даты начала и окончания проекта,
- приоритет проекта (целочисленный).

Дополнительное задание 1:

1) Добавить сущность «Задача», используя Entity Framework Code First DB Migrations.

Функциональные требования:

- возможность создавать/просматривать/редактировать/удалять информацию о задачах;
- возможность добавлять и удалять задачи из проекта (один проект может содержать несколько задач);
- возможность добавлять и менять сотрудников (исполнителей проекта) на задачи (у задачи может быть только один сотрудник, у одного сотрудника может быть много задач);
- для просмотра задач предусмотреть различные способы фильтрации (например, по статусу) и сортировки (по основным полям).

Необходимая для хранения информация:

- название задачи,
- автор задачи (связь с сущностью «Сотрудник»),
- исполнитель задачи (связь с сущностью «Сотрудник»),
- статус задачи (перечисление ToDo/InProgress/Done),
- комментарий,
- приоритет задачи (целочисленный).

2) Используя ASP.NET Identity 2.0, добавить в проект авторизацию и контроль прав доступа.

Функциональные требования:

- добавить три типа ролей: руководитель, менеджер проекта, сотрудник;
- руководитель может видеть все страницы и управлять всеми сущностями;
- менеджер проекта может: а) просматривать свои проекты и назначать/убирать сотрудников с проекта, б) управлять задачами на проекте, назначать исполнителя, менять статус. Менеджер проекта не может добавлять в систему новых сотрудников;
- сотрудник может: а) просматривать свои проекты, б) просматривать свои задачи и менять их статус.

Дополнительное задание 2:

Используя один из JavaScript MVVM фреймворков (например, Knockout.js или Vue.js), реализовать визард создания проекта.

Функциональные требования:

- визард состоит следующих шагов:
  - 1) заполнение названия проекта, дат начала и окончания проекта, приоритета проекта;
  - 2) заполнение компании-заказчика и компании-исполнителя;
  - 3) выбор руководителя проекта из списка всех сотрудников;
  - 4) выбор исполнителей для проекта из списка всех сотрудников;
  - 5) загрузка документов проекта;
- на шагах выбора руководителя и исполнителя проектов использовать выпадающий список сотрудников с возможностью частичного ввода текста для последующего поиска сотрудников на сервере через AJAX-запрос и ASP.NET Web API;
- на пятом шаге визарда реализовать HTML5 file uploader с возможностью drag & drop.