[ВАКАНСИИ](https://itpdevelopment.ru/index.html" \l "job)

Тестовое задание

Создайте приложение учета рабочего времени. После завершения вам необходимо загрузить проект в любой открытый репозиторий GIT.

**Это приложение должно уметь:**

1. Показать пользователю текущие задачи в разрезе проектов и дат
2. Подсчет времени, затраченного на задачу
3. Уметь создавать задачи и проекты

**Для реализации необходимо использовать:**

1. С#. ASP.NET.Core Mvc. Entity Framework.
2. JavaScript/TypeScript. ReactJS/Angular or any other JS framework
3. SCSS/CSS. Bootstrap or any other Framework

**Приложение должно иметь многоуровневую архитектуру:**

1. DAL
2. Buisness
3. UI.

**Пользовательский интерфейс**

1. Таблица задач должна отображать всю необходимую пользователю информацию и иметь следующие столбцы
   * Номер по порядку
   * Название проекта
   * Количество времени, потраченного на задачу в часах и минутах. Формат 00:00
   * Название задачи (ссылка на страницу редактирования)
   * Время начала
   * Время окончания
2. На странице с таблицей задач нужно отобразить общее время, потраченное на задачи из таблицы.
3. Над таблицей нужно отобразить DatePicker для фильтрации задач по дате, а также отобразить выпадающий список проектов, для фильтрации по проекту.
4. Для создания задачи необходимо использовать всплывающий диалог с формой. После сохранения данных страница не должна перезагружаться, а данные таблицы должны обновляться. Поля:
   * Название задачи (обязательный параметр)
   * Проект (обязательный параметр, выпадающий список)
   * Время начала и время окончания. Время начала не может быть больше времени окончания
   * Описание задачи. Это может быть как файл, так и просто текстовое поле, предусматривающее переключение.
5. Задача редактируется на отдельной странице. Отображаемые поля такие же, как и при создании (пункт 2). На этой странице предоставить возможность отображать, добавлять и удалять наборы Описаний текущей задачи.
6. На странице с таблицей задач нужно отобразить общее время, потраченное на задачи из таблицы



**Подсчет времени**

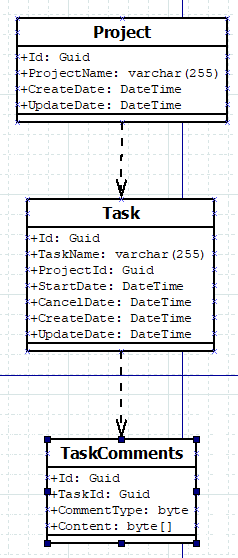
Столбец «Количество затраченного времени на задачу» является динамическим и рассчитывается по формуле:

"Количество времени, затраченное на задачу" = "Время окончания" - "Время начала"

Если время начала указано, а время окончания не указано, то задача считается выполняемой, а время окончания считается от текущего времени. ДатаВремя.Сейчас. Подсчет времени должен быть реализован на сервере.

**Структура базы данных**

Обеспечить заполнение всех таблиц в БД тестовыми данными при обновлении миграций.



[VACANCIES](https://itpdevelopment.ru/index.html#job)

Test Job

Create a time attendance application. After completion, you need to upload the project to any open GIT repository.

**This application should be able to:**

1. Show the user the current tasks in the context of projects and dates
2. Calculate the amount of time spent on a task
3. Be able to create tasks and projects

**For implementation, you need to use:**

1. С#. ASP.NET.Core Mvc. Entity Framework.
2. JavaScript/TypeScript. ReactJS/Angular or any other JS framework
3. SCSS/CSS. Bootstrap or any other Framework

**The application must have a layered architecture:**

1. DAL
2. Buisness
3. UI.

**User Interface**

1. The task table should display all the necessary information for the user and have the following columns
   * Number in order
   * Project name
   * The amount of time spent on a task in hours and minutes. Format 00:00
   * Task name (Link to edit page)
   * Start time
   * End time
2. On the page with the table of tasks, you need to display the total time spent on tasks from the table.
3. Above the table, you need to display the DatePicker to filter tasks by date, as well as display a drop-down list of projects, to filter by project.
4. To create a task, you must use the pop-up dialog with the form. After saving the data, the page should not be reloaded and the table data should be updated. Fields:
   * Task name (required parameter)
   * Project (required parameter, drop-down list)
   * Start time and end time. Start time cannot be greater than End time
   * Description of the task. It can be either a file or just a text field, provide for switching.
5. The task is edited on a separate page. The displayed fields are the same as when creating (item 2). On this page to provide the ability to display, add and delete sets of Descriptions of the current task.
6. On the page with the table of tasks, you need to display the total time spent on tasks from the table



**Time counting**

The column "The amount of time spent on the task" is dynamic and is calculated by the formula:

"Amount of time spent on the task" = "End time" - "Start time"

If the Start time is specified and the End Time is not specified, then the task is considered to be in progress and the end time is considered from the current time. DateTime.Now. Time counting must be implemented on the server.

**Database structure**

Provide filling of all tables in the database with test data when updating migrations.

