Тестовое задание

Результат выкладывайте на GitHub – заодно познакомитесь с ним.

Файлы проекта **.idea** и скомпилированные классы в **output** или **target** выкладывать не надо.

Требуемые технологии:

- Maven (для сборки проекта);
- Tomcat 8 или 9 (для тестирования своего приложения);
- Spring (версия не ниже 4.3.12);
- Hibernate (версия не ниже 5.3);
- MySQL (база данных (БД)). Для упрощения тестирования называйте все свою базу ***test***, с логином и паролем ***root*** (нам не нужно будет для тестирования создавать кучу лишних и ненужных баз);
- Frontend: ***Spring MVC*** *или* ***Angular***.

Версии можно смело брать самые последние. Конфликтов быть не должно.

Приложение должно быть в виде проекта, который собирается с помощью Maven. Обязательно должен присутствовать скрипт для создания и наполнения тестовыми данными вашей базы данных. **Предупреждение**: скрипт на SQL – это **скрипт, а не дамп** базы данных из WorkBench, PhpMyAdmin, и прочих программ. Не ленитесь и напишите скрипт сами. В тестовые данные вставьте от 21 до 40 записей (компьютерных комплектующих).

Скрипт должен выглядеть как-то так:

```
USE test;

DROP TABLE IF EXISTS part;
CREATE TABLE part(
  id INT(11) NOT NULL AUTO_INCREMENT,
  ...
  PRIMARY KEY (id))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8;

INSERT INTO part (...) VALUES (...),
  (...),
  ...
;
```

Если для старта приложения после успешного деплоя нужны какие-то дополнительные действия – опишите все это в каком-нибудь readme в корне проекта.

Визуальную часть проектов не рисую – UX и UI нужно придумать самому.

Огромная просьба: тестируйте свое приложение. Бывало много случаев, когда присылали полностью нерабочее приложение, или с огромным количеством багов.

В посте стажировка вы можете задавать вопросы по тестовому заданию, если, например, непонятно что такое пейджинг, или, как хранить булевы значения в БД. На задания вам дается 3 недели (вполне достаточно если учесть, что они делаются и за день). Если у вас уходит много времени и что-то не получается, то не расстраивайтесь – возможно вы сильно забурились в материал, что совсем не обязательно.

Ниже приводится ссылка на архив с полезными книгами, которые помогут вам в решении тестового задания. Вам не нужно их все перечитать. Используйте как справочники.

литература

Задание: PARTS (компьютерные комплектующие)

Реализовать простенькое приложение Parts-list, для отображения списка деталей для сборки компьютеров на складе. Записи хранить в базе данных. Схему таблички для хранения нужно придумать самому (достаточно одной таблицы).

Нужно показывать **список уже имеющихся деталей** (**с пейджингом** по 10 штук на странице). В списке должно быть наименование детали (String), обязательна ли она для сборки (boolean) и их количество на складе (int). На склад можно **добавлять** новые детали, **редактировать** существующие детали (любое из полей), **удалять**.

- Должна быть **сортировка** по принципу:

все детали, детали, которые необходимы для сборки, опциональные детали.

- Должен быть **поиск** по наименованию детали.

Бизнес-требование: ниже списка деталей всегда выводить поле, в котором выводится, сколько компьютеров можно собрать из деталей в наличии. Для сборки одного компьютера должны быть в наличии все детали, которые отмечены как необходимые.

Пример 1:

Наименование	**Необходимость**	**Количество**
материнская плата	да	3
звуковая карта	нет	2
процессор	да	2
подсветка корпуса	нет	0
HDD диск	нет	1
корпус	да	10
память	да	10
SSD диск	да	15
видеокарта	нет	7

Можно собрать	2	компьютеров

Пример 2:

Наименование	**Необходимость**	**Количество**
материнская плата	да	0
звуковая карта	нет	2
процессор	да	2
подсветка корпуса	нет	100
корпус	да	10
память	да	10
SSD диск	да	15
видеокарта	нет	7

Можно собрать	0	компьютеров