**Перший рівень — відпрацюй навички на базовому рівні.**

1. В баг-трекінговій системі (Jira) опиши 3 баги, які знайдеш на сайті https://www.headhunterhairstyling.com/

https://beetrootun.atlassian.net/browse/BEET-27

**Другий рівень — детальніше заглибся в практику.**

1. Виконай завдання попереднього рівня.

2. Наведи власні приклади багів, які можуть мати такі комбінації:

Severity - Critical/Priority – Low

* Имейл не отправляется, если во введенном адресе есть символ ~. Это критический баг, тк важная функция (отправка имейла) не отрабатывает, но приоритет низкий, тк мало вероятности в том, что в ближайшее время кто-то будет отправлять имейл на такой адрес, потому что такие адреса – редкость.
* По нажатию на кнопку «Оформить заказ» в футере магазина не происходит переход на страницу оформления заказа. Баг критичен, потому что не отрабатывает ключевая функция, однако, в то же время большинство покупателей переходят к оформлению заказа из корзины, а не из футера, поэтому можно фиксить баг немного позже.

Severity – Minor / Priority – Highest

* Лого на главной странице сайте отображается сверху на главном меню (поехала верстка). Баг не критичен, тк никакая функция не задета, но исправлять надо срочно, потому что это первое, что бросится в глаза посетителю сайта и испортит все впечатление от пребывания там.
* Поле подтверждения пароля не обозначено красной звездочкой как обязательное. Баг не критичен, тк функция написано верно, и все равно сработает валидация, если поле будет пустым. Но фиксить надо срочно, потому что этот дефект может негативно влиять на желание пользователя продолжать регистрацию.

**Третій рівень — різнобічно опануй тематику уроку.**

1. Виконай завдання двох попередніх рівнів.

2. Продовжуємо розвивати стартап для застосунку, який дозволяє обмінюватися фотографіями котиків. Команда тестувальників скаржилась, що без баг-трекінгової системи все було дуже погано. Тому ви вирішили запровадити систему відстежування багів Jira. Ти як керівник/ця маєш створити життєвий цикл багу. Які статуси туди будуть входити? В якій послідовності? Чому саме такі?

В такой ситуации я бы взяла жизненный цикл бага из своего же опыта. Такая модель показала себя отличным вариантом для группы из 2-3 девов и 1-2 qa. Небольшое количество человек дает возможность легко коммуницировать вне баг-трекинговой системы, поэтому сложная модель в принципе и не нужна. Итак, при создании баг-репорта, issue падает в статус Open. Assigned Dev, беря ее в работу, меняет статус на In Progress. Это удобно для менеджера или лида, чтобы понимать, в каком состоянии находится разработка и багфиксинг на данный момент, а также корректно оценивать загрузку каждого из членов команды. После фикса дев переводит issue в статус Resolved и проставляет в Assignee кого-то из QA, обычно того, кто завел баг. Проверяя фикс, QA не меняет статуса issue, тк с большой долей вероятности баг пофикшен качественно, а значит issue можно условно считать почти закрытым. Именно поэтому QA не ставит статус In Progress. Из Resolved qa переводит issue либо в Closed (fixed). Либо в Reopen, снова назначая дева для дальнейшего исправления. В таком случае дев снова ставит In Progress , и все идет по кругу. Жизненный цикл бага считается оконченным, когда ему присвоили статус Closed (fixed).