

Скачайте проект с <https://assets.codeforces.com/files/b3aa7eec048205c7/lesson9.7z>
Запустите проект (поправьте src/main/resources/application.properties). Используйте базу данных вида uXX. Запуск создаст таблицу user. Добавьте к ней колонку passwordSha, которая по-умолчанию NULL. Убедитесь, что всё работает - регистрация+вход+выход, просмотр пользователей. Создайте нескольких пользователей, дайте им разные права (просто руками добавьте записи в user_role). Полезный справочный ресурс https://en.wikibooks.org/wiki/Java_Persistence/Relationships

1.

Добавьте сущность простейшую сущность Comment с четырьмя полями id, text (используйте аннотацию @Lob), user, post. Добавьте и List<Comment> comments в Post. Настройте двунаправленный мэппинг many-to-one и one-to-many для отношения комментарий-пост и однонаправленный мэппинг many-to-one для отношения комментарий-пользователь (то есть коммент знает своего пользователя, а в пользователе комментарии не хранятся). Сделайте простейшую страницу с отображением одного Post и списка его комментариев под ним (по urlу /post/{id}). Перед списком комментариев для залогинённого пользователя добавьте форму добавления нового комментария (просто одна textarea и кнопка). Пусть клик по заголовку одного Post теперь ведёт на страницу индивидуального Post (например, на главной будут все посты с ссылками в заголовках).

2.

Добавьте сущность Tag (тег). У него есть только id и name. Теги должны находиться с Post в отношении многие-ко-многим. Сделайте однонаправленный мэппинг многие-ко-многим, то есть поле Post#tags. Надо делать по образу и подобию как связаны пользователи и роли. Добавьте руками несколько тегов (записи в таблицу tag) и несколько тегов к posts (записи в таблицу post_tag). Отображайте список тегов при отображении поста.

3.

Добавьте поле Tags в форму создания поста. Считайте, что туда надо ввести список тегов через пробельные символы (whitespaces). Выбрасывайте ошибку валидации, если ввели тег (после разбиения на слова), что не является простым словом из латинских букв. При сохранении нового Post надо сохранить его вместе с тегами (возможно, в таблице tag появятся новые записи, возможно будут использованы старые).