

Сдача работы и баллы

Работу необходимо будет сдать лично. На своей паре или в доп. время. Не тяните, распределяйте время, чтобы не было столпотворений!

При проверке будет оцениваться соответствие всем нижеперечисленным требованиям, реализация правильной архитектуры, умение применять используемые технологии, удобство пользования системой, защищенность системы.

Сайт должен быть задеплоен на общедоступный сервер.

Работа без выполнения списка обязательных требований ниже не принимается. Невыполнение каждого дополнительного требования к наполнению/реализации уменьшает максимальные баллы на 20%.

Обязательные требования

Сайт, написанный с использованием Spring MVC Framework.

У сайта должна быть идея, обладающая новизной и актуальностью. Если сомневаетесь в сложности и новизне проекта, лучше подойдите-уточните. Либо же можно воспользоваться своей предыдущей семестровой и оптимизировать её.

Сайт должен иметь удобный интерфейс.

Необходимо использовать HTML5, CSS3. CSS-препроцессоры не запрещаются, но нужно знать их основные принципы. Если используете фреймворки (Bootstrap и проч.), должны уметь пояснить, что за классы используете, особенно классы сетки.

Необходимо корректно работать с формами: проверка данных, защита от повторной отправки там, где это нужно.

Нужно использовать любой шаблонизатор, *можно JSP*. В коде видов нельзя напрямую вызывать Java-код. Должна быть развитая система шаблонов (свои теги или наследование шаблонов, разделение видов на несколько частей).

Сайт должен иметь аутентификацию, авторизацию, регистрацию.

Сайт должен хранить данные в реляционной СУБД: MySQL, MSSQL, MariaDB или PostgreSQL. Можно ли использовать другие СУБД нужно уточнить у преподавателя.

Для работы с СУБД необходимо использовать и Spring Data JPA, и JPA. То есть должны быть как и репозитории от Spring, так и самостоятельно написанные запросы с помощью JPQL/CriteriaBuilder.

Система должна работать, как минимум с пятью JPA сущностями. Должны быть связи M2M, O2M.

Как минимум, для одной из сущностей должен быть реализован полностью CRUD: не обязательно в виде админки, но должны быть некие экшены контроллеров, которые вызывают эти методы.

Система должна быть построена по модели MVC, и следует придерживаться логики Controller->Service->Repository(DAO) и тд. Необходимо придерживаться принципов SOLID. Весь код должен быть корректно поделен на пакеты. Рекомендуется использовать DTO-, Request-, Forms-классы, где это уменьшает связность, абстрагирует архитектуру.

Проект должен быть основан на Spring Boot: нужно использовать стартеры, настройки и запускающий класс.

Все реализации требований должны вписываться в бизнес-логику и выполнение соответствующего кода должно быть возможно из-за действий пользователя.

Требования к наполнению, наличию реализаций

- Необходимо применить Javascript+jQuery. Нужно использовать AJAX.
- Все исключения, проблемы должны корректно обрабатываться: обычные экшены – приводить к отдаче самостоятельно оформленной веб-страницы, а AJAX-запросы получить JSON-ответ с информацией о проблеме. Не должно быть дефолтных страниц с ошибками от сервера приложений или Spring Boot.
- Использование стороннего API, не используя специальную библиотеку для этого API, а самостоятельное описание запросов на уровне HTTP (например, библиотека okhttp). Примеры: вход или восстановление пароля через отправку смс, использование курсов валют для пересчета стоимости корзины. Наличие простого информера для сайта типа вывода погоды не подходит!

- Использование Spring Security. То есть полноценная работа с пользователем: закрытая часть сайта, аутентификация, регистрация.
- Организация REST API хотя бы для одной сущности.
- Использование конвертеров, в том числе и своих.
- Логирование для обработки исключений.
- Наличие 2 нестандартных методов для репозитория в Spring: самостоятельно написанных методов с использованием @Query, CriteriaBuilder, но не повторяющих функционал методов, описываемых в Spring Data JPA.
- Один запрос с подзапросом.

Сроки

Дедлайн досрочной сдачи: 11 мая. Можно получить +4 балла баллов сверх баллов семестровой работы при строгом выполнении всех требований.

Дедлайн сдачи: 18 мая.

Разбалловка

При несоответствии каждому дополнительному требованию максимальные баллы уменьшаются на 20%.

Если обнаружено, что какой-то участок кода списан, то баллы списавшего и давшего списать, делятся пополам.

Все уменьшения баллов в разы целочисленные с округлением в меньшую сторону.

Фронтенд (вёрстка, дизайн, шаблоны, JS) – 3

Работа с базой (репозитории, 4 сущности, M2M, O2M, свои запросы) – 4

Архитектура (слои абстракций, взаимодействие модулей, шаблоны проектирования) – 4

Общие правила построения сайтов (упоминание контекста, защита от повторной отправки, проверки ввода, корректный вывод, обработка запроса и проч) – 6

Использование продвинутых техник/технологий, в т.ч. Spring (логирование, правильное описание конфигураций, настройка

окружения, конвертеров, централизованная обработка исключений и т.д.) – 7

Корректное использование API – 1

Итого: 25 баллов

Пример: студент сдал работу без 2 дополнительных требований и у него было очень много проблем с фронтендом. 25-3 балла=22 балла. И -40% -> 13 баллов.

Доп.задание:

Подключить OAuth-сервис без использование библиотеки, которая непосредственно решает эту задачу. Максимум: библиотека для отправки http-запросов, генерации секретных кодов. Нужно реализовать функционал “войти через”.

+4 балла (но не более 25 суммарно)