Создание web-интерфейса для проекта SQL2 на языке Java.

Contents

Chapter I

Introduction

В данном проекте Вам предстоит реализовать web-интерфейс для проекта SQL2. Вам нужно будет разработать web-приложение с использование MVC-фреймворка. Приложение должно поддерживать осуществление CRUD-операций, импорт/экспорт таблиц, осуществление разработанных на прошлом шаге операций/функций через графический интерфейс, а также логирование действий пользователя.

Chapter II

Rules

- Представление модели базы данных, а также процедур, связанных с добавлением/удалением/корректировкой данных находятся в проекте SQL2
- Для оценки ваше решение должно находится в git-репозитории
- У вас есть вопрос? Спросите своего соседа справа. В противном случае попробуйте обратиться к своему соседу слева
- Ваше справочное руководство: коллеги / Internet / Google

Information

Паттерт MVC

Паттерн MVC (Model-View-Controller, Модель-Представление-Контроллер) представляет из себя схему разделения модулей приложения на три отдельных макрокомпонента: модель, содержащую в себе бизнес-логику, представление - форму пользовательского интерфейса для осуществления взаимодействия с программой и контроллер, осуществляющий модификацию модели по действию пользователя.

Модель хранит и осуществляет доступ к основным данным, производит по запросам операции, определенные бизнес-логикой программы, то есть отвечает за ту часть программы, которая отвечает за все алгоритмы и процессы обработки информации.

Контроллер выполняет функцию связующего элемента между интерфейсом и моделью, осуществляет модификацию модели. Через него формируются запросы на изменение модели.

Представление показывает пользователю данные из модели в удобном и понятном виде, интерфейс программы. В идеале в представлении не должно быть никакой бизнес-логики.

Server-Side Rendering (SSR)

Серверный рендеринг (server-side rendering, SSR) — это технология рендеринга приложения или сайта на сервере, а не в клиентском браузере. При серверном рендеринге в ответ на запрос на сервере генерируется весь HTML-код страницы. Это исключает необходимость дополнительных запросов данных со стороны клиента, так как сервер берёт всю работу на себя, прежде чем отправить ответ. Главное преимущество SSR — возможность повысить производительность приложения.

Упрощенно SSR работает следующим образом:

1. Браузер запрашивает страницу;
2. Сервер генерирует HTML-страницу для вывода и отправляет ее обратно;
3. Браузер отображает HTML;

Chapter III

Part 1. Реализация web-приложения для проекта SQL2

Необходимо реализовать web-приложение для проекта SQL2

Общие требования

- Программа должна быть разработана на языке Java версии 8
- Код программы должен находиться в папке src
- При написании кода необходимо придерживаться Google Style
- Необходимо разработать Web-приложение
- Программа должна быть реализована с использованием любого MVC-фреймворка (Spring)
- Программа должна быть реализована с использованием паттерна **MVC**, а также:
  - не должно быть кода бизнес-логики в коде представлений;
  - не должно быть кода интерфейса в контроллере и в модели;
  - контроллеры должны быть тонкими;
- Рендеринг страниц осуществлять на стороне сервера (использование технологии **Server-Side Rendering**)
- Вам необходимо полностью переиспользовать базу данных из проекта SQL2, включив её в компонент Model

- В качестве референса по дизайну вы можете использовать бренд-бук Школы или визуальный стиль платформы (см. materials)
- Дизайн приложения должен быть интуитивно-понятным

Требования к содержанию

- Главная страница должна содержать:
    - Навигационное меню, обеспечивающее переход к основным разделам приложения: *«Данные»* и *«Операции»*
    - Поле *«О себе»*, содержащее основную информацию о студенте, выполнившем проект
- Графическая оболочка страниц *«Данные»* и *«Операции»* должна содержать следующие разделы:
    - Шапка, по нажатию на которую можно осуществить переход на главную страницу
    - Навигационное меню, позволяющее осуществить переход по основным разделам
    - Основную часть раздела: содержательный информационный текст, иллюстрации и т.п
    - Навигационная панель, осуществляющая передвижение по подразделам выбранного раздела (в случае необходимости)
- Раздел *«Данные»* должен содержать подразделы, которые позволяют через GUI поддерживать следующий функционал:
    - Совершать CRUD-операции по всем таблицам
    - При любой модификации таблиц (create, update, delete) приложение должно запрашивать у пользователя подтверждение на осуществление операции
    - После любого вида модификации таблиц пользователю должна выводиться измененная таблица
    - Импорт и экспорт данных для каждой таблицы из файлов/в файлы с расширением *.csv*
- Раздел *«Операции»* должен содержать компоненты:
    - Блок, содержащий все возможные для вызова запросы из проекта SQL2, наименование/краткое описание сути запроса
    - Блок с возможностью самостоятельного ввода SQL-запроса пользователем

- Раздел *«Операции»* должен содержать подразделы, которые позволяют через GUI поддерживать следующий функционал:
    - Выбор желаемой процедуры / функции / запроса из разработанных в проекте SQL2 с выводом результата и возможность экспорта результата в файл разрешения .csv
    - В случае необходимости введения параметров для выполнения процедуры или функции, графический интерфейс должен предоставлять форму для ввода данных
    - Если введенные аргументы/SQL-запрос были некорректны, то приложение должно обрабатывать подобную ситуацию (выдавать ошибку о некорректности введенных и данных и предлагать повторную попытку ввода)
    - При осуществлении процедур / функций / запросов необходимо вызывать оригинальные хранимые операции, описанные в базе данных на языке SQL
- Конфигурация приложения должна осуществляться при помощи файла конфигурации, который включает в себя строку подключения к СУБД.

Part 2. Логирование

Необходимо реализовать логирование всех действий пользователя (файлы логов записывать в папку logs). Каждый день создается новый файл логов. Название файлов должно соответствовать шаблону *logs_dd-MM-yy-hh-mm-ss*

У каждой записи должен быть обозначен ее уровень важности:

- **Info**: ожидаемое событие;
- **Warning**: неожиданные события, которые позволяют продолжить работу приложения;
- **Error**: событие, которое не позволяет дальнейшую работу программы;

Part 3. Дополнительно. Развертывание приложения

Подготовить приложение к запуску. Для этого необходимо упаковать в docker-контейнеры:

- базу данных
- проксирующий сервер (использовать nginx)
- web приложение

Подготовить docker-compose для запуска всего приложения. При этом "наружу" должен смотреть только docker-контейнер, содержащий Nginx.

Give feedback