Цель: Изучить различные библиотеки и фреймворки создания серверной части веб-приложений.

Задание: В ходе выполнения работы вам необходимо 3 раза с использованием различных технологий создать серверную часть веб приложения.

Требование к серверной части:

1. Целостность (система должена предназначаться для выполнения какой-либо одной функции);
2. HTTP API интерфейс (доступ ко всем функциям должен быть через API);
3. Подключение к базе данных;
4. Наличие функции входа в систему и разграничения ролей;
5. Наличие функции загрузки и скачивания файлов (например, аватарок пользователей).

Особые требования: вам необходимо реализовать одно и то же приложение 3 раза с использованием различных технологий. Получившиеся системы должны реализовывать одинаковое поведение на различных версиях. При освоении технологии попытайтесь изучить принципы и паттерны построения приложений, которые применяются при её использовании.

Технолоигии: при выполнении задания вы должны выбрать две технологии из следующих трех: ExpressJS, Ruby on Rails, Django. Третью технологию вы можете выбрать на свое усмотрение из следующего списка: Flask, Laravel, Fiber, CakePHP, Play for Scala, Koa.js и другие (также вы можете выбирать Java Spring и ASP.NET, но только в том случае, если вы раньше с ними не сталкивались и никогда с ними не работали!!!).

Требования к отчету: В отчете необходимо показать всю проделанную работу, а также сравнить технологии, выделить их плюсы и минусы, сделать вывод под какие задачи подходит каждая из технологий.

Контрольные вопросы:

1. Трехуровневая архитектура веб-приложения. Основные звенья.
2. Монолитная и микросервесная архитектуры. Основные отличия.
3. Проектирование API. REST API. RPC API. Схемы. Отличия.
4. Протокол HTTP. Основные механизмы.
5. Протокол SOAP. Особенности. Файл WSDL. Структура файла WSDL.
6. Протокол GraphQL. Особенности. Запросы (query) и мутации (mutation).
7. Платформа NodeJS. Особенности. Преимущества и недостатки.
8. Фреймворк express. Особенности. Преимущества и недостатки.
9. Язык программирования Ruby. Особенности. Преимущества и недостатки.
10. Фреймворк Ruby on Rails. Особенности. Преимущества и недостатки.
11. Фреймворк Django. Особенности. Преимущества и недостатки.
12. Схема MVC. Компоненты. Особенности использования.