Сучасні методології і технології розробки програмного забезпечення

CMiТРПЗ-3

# Розділ 1. Основи сумісної розробки

## 1.1. Огляд систем контролю версій

1.1.1. Принцип роботи систем контролю версій.

1.1.2. github :: https://github.com/

https://git-scm.com/book/uk/v2

1.1.2.1. Github як сховище даних.

1.1.2.2. Основні команди: git init, git clone, git status, git diff, git log, git checkout

1.1.2.3. Команди: git pull (git fetch и git merge), git add, git push, git commit

1.1.2.4. Цикл розробки та філософія роботи з git

1.1.3. bitbucket :: https://bitbucket.org/

https://itpro.ua/product/atlassian-bitbucket/?tab=description

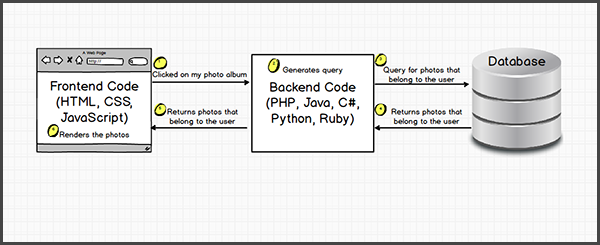
http://s-engineer.ru/pamyatka-po-rabote-s-git-bitbucket/

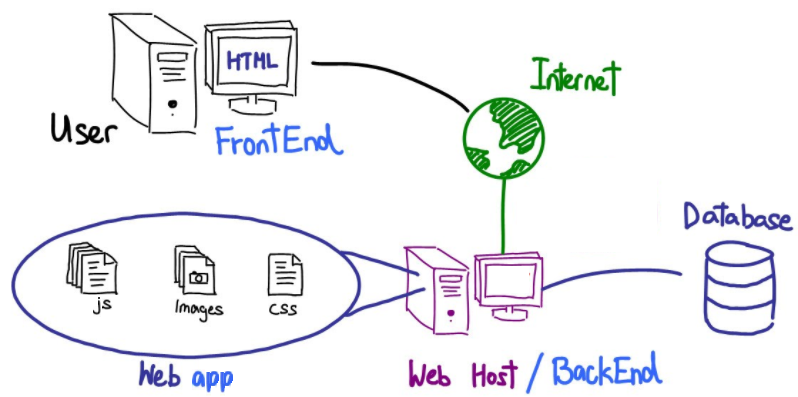
https://eax.me/github-vs-bitbucket/

1.1.4. Microsoft Team Foundation Server

# Розділ 2. BackEnd development

## 2.1. Основи веб-розробки

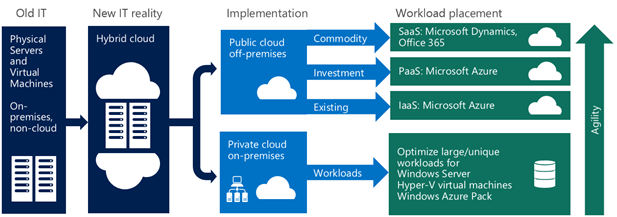




2.1.1. Поняття WebServer

2.1.1.1. Налаштування локального веб-серверу та запуск додатку

2.1.1.2. Хмарні платформи та служби



2.1.3. Формат передачі даних від сервера JSON vc XML

2.1.4. Структура URL, використання для передачі параметрів

2.1.5. Використання Cookies

## 2.3. Основні задачі бекенд-розробки



## 2.4. Огляд інструментів розробки Microsoft

2.4.1. MS SQL Server

2.4.2. ORM Entity Framework

2.4.3.WebAPI Application

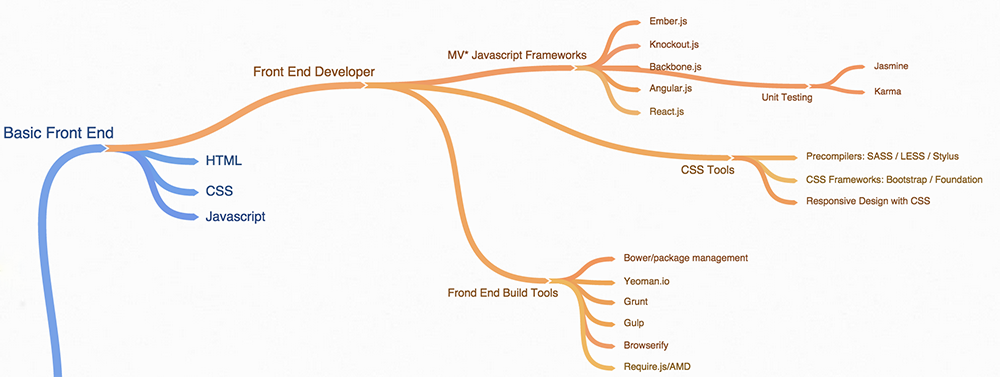
## 2.4. RESTful Architecture

2.4.1. API Development

2.4.2. Routing

# Розділ 3. FrontEnd development

## 3.1. Основні задачі фронтенд-розробки



## 3.1. Принципи роботи браузера

## 3.2. HTML&CSS, HTML 5, CSS 3

3.2.1. W3C Standards. Основні теги HTML, HTML5

https://www.w3schools.com/

3.2.2. Синтаксис CSS, CSS 3

3.2.3.1. Основні контейнери CSS

3.2.3.1. Оптимізація графіки з використанням спрайтів

3.2.3.1. Робота зі шарами (layer)

3.2.3.1. Тіні, градієнти, 2D та 3D трансформація в CSS3

3.2.3. Елементи управління на сторінці

3.2.4. SEO оптимізація



## 3.3. JavaScript

http://javascriptissexy.com/how-to-learn-javascript-properly/

https://github.com/HowProgrammingWorks/Dictionary

3.3.1. Data Types, Control Flow, and Loops

3.3.2. [Objec](https://github.com/HowProgrammingWorks/DataTypes)t, array

3.3.3. Functions, [Callback](https://github.com/HowProgrammingWorks/Callbacks)

3.3.4. Обробка подій в моделі DOM

3.3.4.1. addEventListener()

3.3.4.2. removeEventListener()

3.3.4.3. stopPropagation()

## 3.4. Відображення даних сервера

3.4.1. Технологія AJAX та робота з HTTP протоколами

3.4.2. Використання XMLHttpRequest

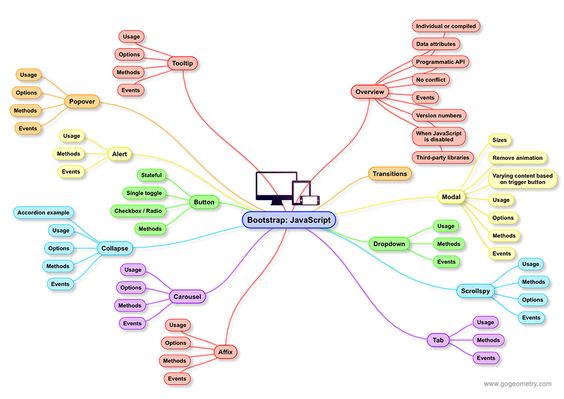
3.4.3. Робота з Cookies: доступ, збереження та читання

3.4.4. Використання LocalStorage

## 3.5. Інструменти фронтэнд

3.5.1. Використання препроцесорів на прикладі LESS

3.5.2. Адаптивність та оптимізація верстки з використанням BootStrap



*3.5.3. Angular 4.0*

## 3.6. Збірка фронтенд проекту

3.6.1. Мета мініфікації й обфускації JS і CSS

3.6.2. Мета мініфікації й обфускації JS і CSS

3.6.3. Огляд готових плагінів для реалізації задач фронтенда

3.6.3.1. Сортування, фільтрація, paging(Select2)

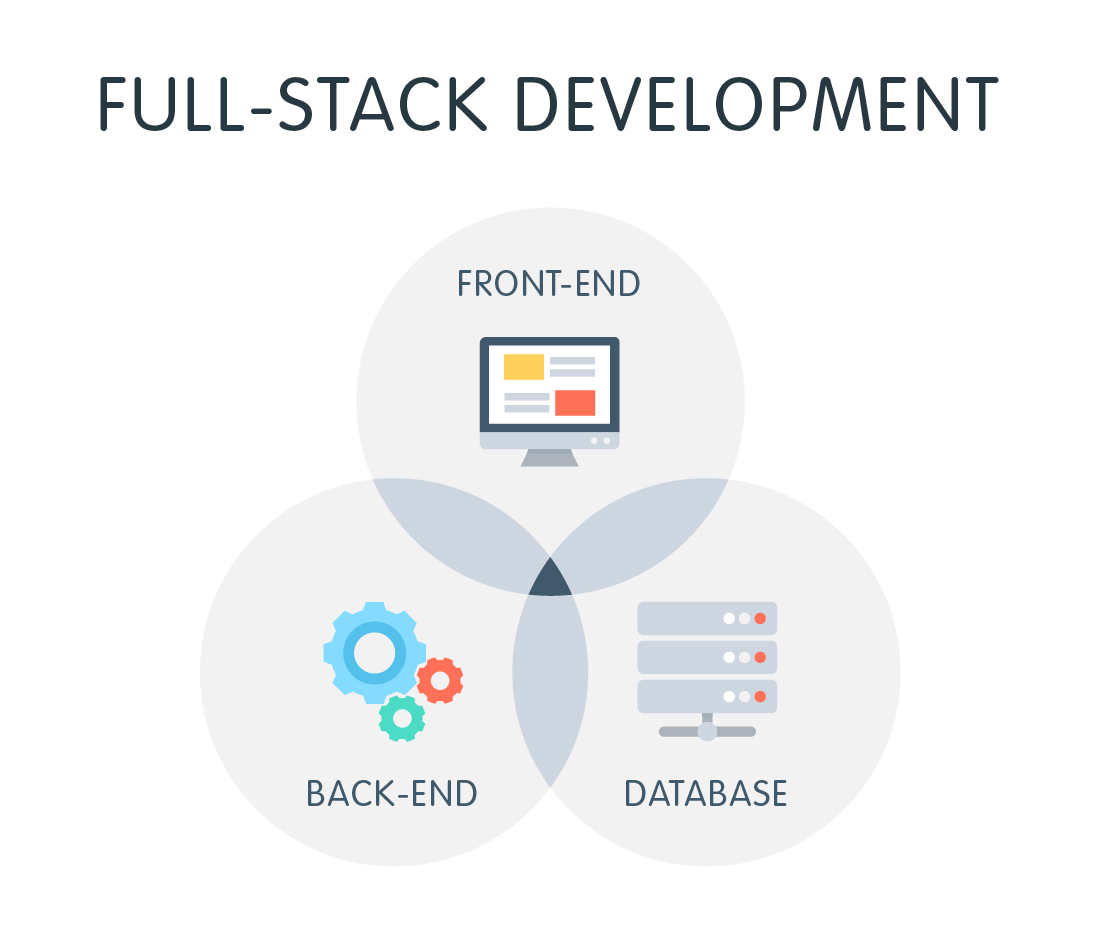
3.6.3.2. Табулятори (FancyBox, BootStrap)

3.6.3.3. Робота з датами (Moment.js, DatePicker)

3.6.3.4. Візуалізація даних (Highcharts)

## 3.7. Модульний підхід до розробки Asynchronous Module Definition (AMD)

# Комп'ютерний практикум



## Завдання 1. Побудувати і реалізувати БД

Основні вимоги: 7- 12 таблиць, цілісність даних, наявність представлень, процедур, які зберігаються

## Завдання 2. Реалізувати backend

Основні вимоги: WebAPI Application, RESTful Architecture, розгорнуто на віддаленому сервері

## Завдання 3. Реалізувати frontend

Основні вимоги:

1. Розробити JavaScript - компонент для відображення даних вихідного JSON файлу, використовуючи LocalStorage

2. Розробити карту фронтенд проекту та зверстати його з використанням BootStrap, реалізувати рядок навігації по сторінках

3. Реалізувати сортування даних в таблиці по зростанню/спаданню. Реалізувати фільтр даних по одному з заданих критеріїв

4. JSON заповнити даними з серверу

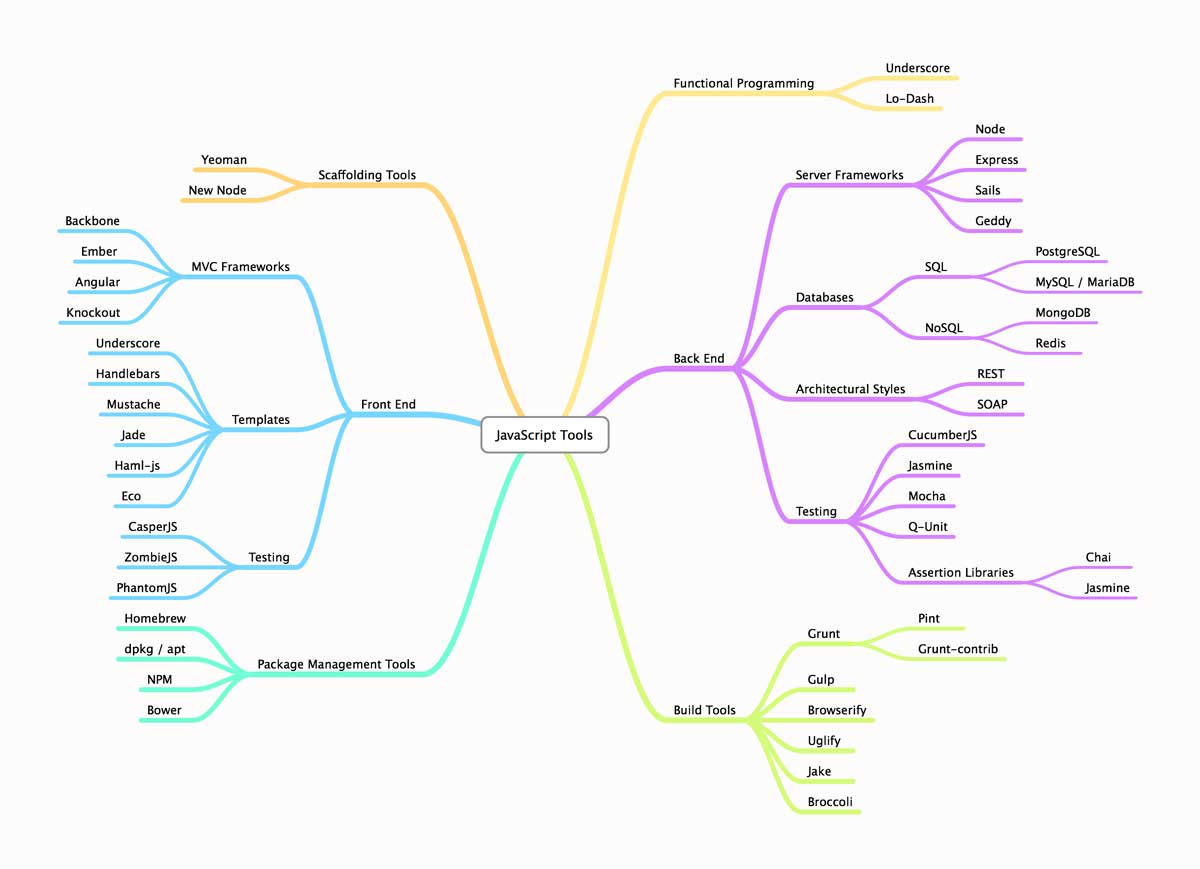
# Рейтингова система оцінювання

Передбачено виконання трьох частин комп'ютерного практикуму з метою навчання. Кожна з частин оцінюється в 25 балів. Максимальна кількість балів в такому випадку: 25\*3=75балів.

Щоб отримати максимально допустиму кількість балів у розмірі 100, пропонується виконувати запропонований об'єм практикуму на власному прикладі.

Додаток 1.

JavaScript Full-Stack Development Roagmap



https://github.com/ivanoats/Full-Stack-JavaScript-Engineering/blob/master/setup/ToolsOverview.md