

**Virtual classroom website**



**Supervisor:**

dr. Kamal Al-Sallom

**INTRODUCED BY STUDENTS:**

Elias Awess

Hassan Eid

Ammar Shalab Al-Sham

Mostafa Audah

Year: 2019/2020

# **Project Objectives:**

## **Introduction:**

Virtual Classroom is an online teaching and learning environment where teachers and students can share and interact with course materials.

Our project is a simple virtual classroom website that help teacher to create course and share course materials with their students.

There are 2 objectives of this project: communication and organization.

## **Commuincation:**

Mean teachers and students can communicate with each other to provide comfortable environment for study, so student can ask questions and answer teacher’s question during each lesson.

## **Oroganization:**

Mean teacher can organize course materials to provide reliable reference to course subject.

Teacher can also share support files and links for more information.

# **Design Pattern: MVC**

Stands for "Model-View-Controller".

MVC is an application design model comprised of three interconnected parts. They include the model (data), the view (user interface), and the controller (processes that handle input).

The MVC model or "pattern" is commonly used for developing modern user interfaces. It provides the fundamental pieces for designing a program for desktop or mobile, as well as web applications.   
It works well with object-oriented programming, since the different models, views, and controllers can be treated as objects and reused within an application.

Below is a description of each aspect of MVC:

1. Model

A model is data used by a program. This may be a database, file, or a simple object, such as an icon or a character in a video game.  
so we define a part of the project to deal with database and file system and other parts request data and files from it.

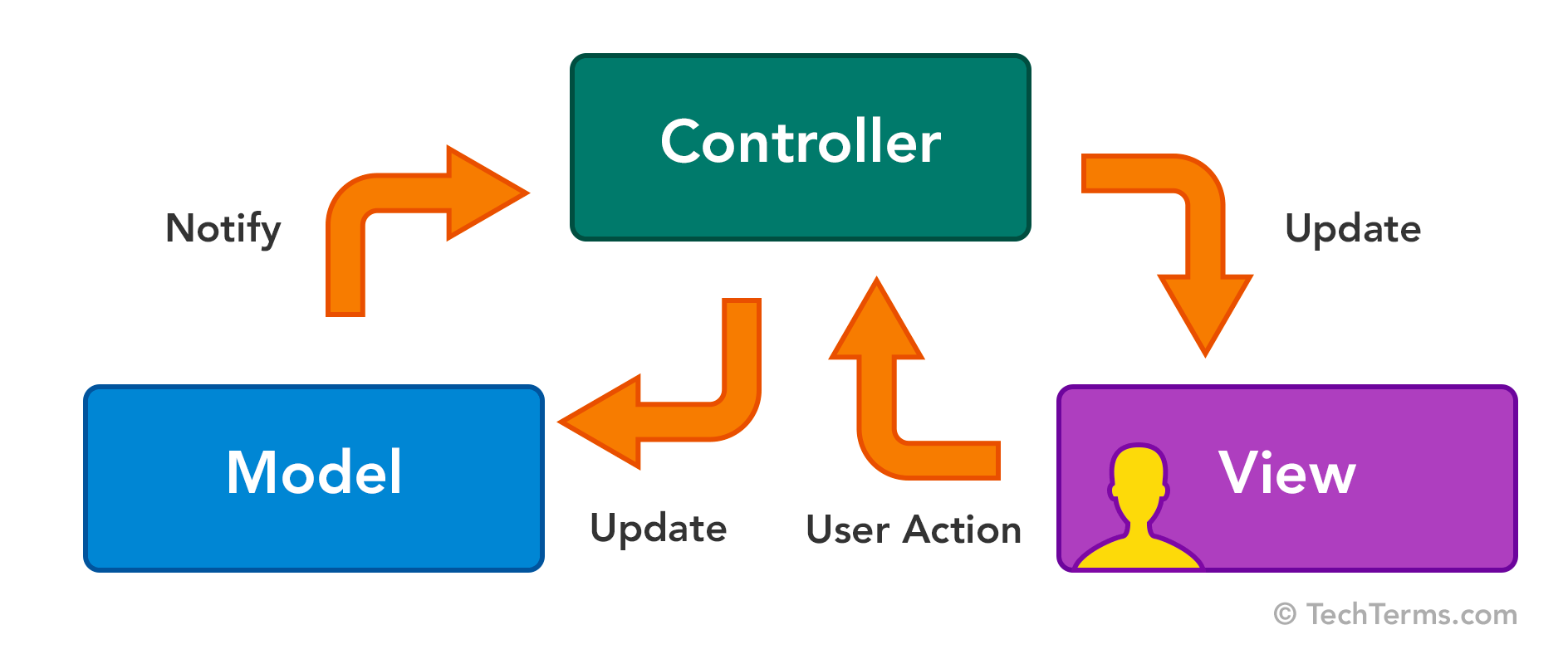
2. View

A view is the means of displaying objects within an application. Examples include displaying a window or buttons or text within a window. It includes anything that the user can see.

3. Controller

A controller updates both models and views.   
It accepts input and performs the corresponding update.   
For example, a controller can update a model by changing the attributes of a character in a video game. It may modify the view by displaying the updated character in the game.   
the controller is considered as a mean between view and model.

The three parts of MVC are interconnected.   
The view displays the model for the user.   
The controller accepts user input and updates the model and view accordingly.   
While MVC is not required in application design, many programming languages and IDEs support the MVC architecture, making it a common choice for developers.



Reference: <https://techterms.com/definition/mvc>

# **Development Tools:**

## **MEAN STACK Web Development:**

**What is MEAN?**

MEAN is a full-stack JavaScript solution that helps you build fast, robust, and maintainable production web applications using MongoDB, Express, Angular, and Node.js.

**Why MEAN?**

MEAN.JS will help you getting started and avoid useless grunt work and common pitfalls, while keeping your application organized.

## **MongoDB: (NoSQL Database)**

Not Only SQL Database

MongoDB is a general purpose, document-based, distributed database built for modern application developers and for the cloud era.

**MongoDB Features:**

1. As a programmer, you think in objects.   
   Now your database does too.  
   MongoDB is a document database, which means it stores data in JSON-like documents.
2. Rich JSON Documents  
   The most natural and productive way to work with data.  
   Supports arrays and nested objects as values.  
   Allows for flexible and dynamic schemas.
3. Powerful query language  
   Rich and expressive query language that allows you to filter and sort by any field, no matter how nested it may be within a document.  
   Support for aggregations and other modern use-cases such as geo-based search, graph search, and text search.  
   Queries are themselves JSON, and thus easily composable.   
   No more concatenating strings to dynamically generate SQL queries.
4. All the power of a relational database, and more...  
   Distributed multi-document ACID (atomicity, consistency, isolation, durability) transactions with snapshot isolation (all reads made in a transaction will see a consistent snapshot of the database).  
   Support for joins in queries.  
   Two types of relationships instead of one: reference and embedded.

Reference: <https://www.mongodb.com>

## **Express JS:**

ExpressJS is Fast, unopinionated, minimalist web framework for Node.js

It used for building Web Applications and APIs.

For Web Application:

Express is a minimal and flexible Node.js web application framework that provides a robust set of features for web and mobile applications.

For API:

With a myriad of HTTP utility methods and middleware at your disposal, creating a robust API is quick and easy.

ExpressJS provides mechanisms to:

* Write handlers for requests with different HTTP verbs at different URL paths (routes).
* Integrate with "view" rendering engines in order to generate responses by inserting data into templates.
* Set common web application settings like the port to use for connecting, and the location of templates that are used for rendering the response.
* Add additional request processing "middleware" at any point within the request handling pipeline.
* Use third-party libraries to add more features to your apps.

Reference: <https://expressjs.com/>

## **Angular:**

Angular is a platform and framework for building single-page client applications using HTML and TypeScript.   
It implements core and optional functionality as a set of TypeScript libraries that you import into your apps.

**Angular Features:**

1. Develop Across All Platforms:

Learn one way to build applications with Angular and reuse your code and abilities to build apps for any deployment target. For web, mobile web, native mobile and native desktop.

1. Speed & Performance:

Achieve the maximum speed possible on the Web Platform today, and take it further, via Web Workers and server-side rendering.  
Angular puts you in control over scalability.

1. Incredible Tooling:

Build features quickly with simple, declarative templates.   
Extend the template language with your own components and use a wide array of existing components.   
Get immediate Angular-specific help and feedback with nearly every IDE and editor.   
All this comes together so you can focus on building amazing apps rather than trying to make the code work.

Reference: <https://angular.io>

## **NodeJS:**

Node.js is an asynchronous event-driven JavaScript runtime.   
Node.js used to run javascript outside the browser.  
Node.js is designed to build scalable network applications.

In Node.js many connections can be handled concurrently.   
Upon each connection, the callback is fired, but if there is no work to be done, Node.js will sleep.  
This is in contrast to today's more common concurrency model, in which OS threads are employed. Thread-based networking is relatively inefficient and very difficult to use. Furthermore, users of Node.js are free from worries of dead-locking the process, since there are no locks. Almost no function in Node.js directly performs I/O, so the process never blocks. Because nothing blocks, scalable systems are very reasonable to develop in Node.js.

Node.js is similar in design to, and influenced by, systems like Ruby's Event Machine and Python's Twisted. Node.js takes the event model a bit further. It presents an event loop as a runtime construct instead of as a library. In other systems, there is always a blocking call to start the event-loop. Typically, behavior is defined through callbacks at the beginning of a script, and at the end a server is started through a blocking call like EventMachine::run(). In Node.js, there is no such   
start-the-event-loop call. Node.js simply enters the event loop after executing the input script. Node.js exits the event loop when there are no more callbacks to perform. This behavior is like browser JavaScript — the event loop is hidden from the user.

HTTP is a first-class citizen in Node.js, designed with streaming and low latency in mind. This makes Node.js well suited for the foundation of a web library or framework.

Node.js being designed without threads doesn't mean you can't take advantage of multiple cores in your environment.   
Child processes can be spawned by using our child\_process.fork() API, and are designed to be easy to communicate with.   
Built upon that same interface is the cluster module, which allows you to share sockets between processes to enable load balancing over your cores.

Reference: <https://nodejs.org/en/about/>

# **Academic Research: SEO**

## **What is Search Engine Optimization (SEO)?**

SEO stands for Search Engine Optimization, which is the practice of increasing the quantity and quality of traffic to your website through organic search engine results.

To understand the true meaning of SEO, let's break that definition down and look at the parts:

1. Quality of traffic:   
   You can attract all the visitors in the world, but if they're coming to your site because Google tells them you're a resource for Apple computers while you're really a farmer selling apples, that is not quality traffic. Instead you want to attract visitors who are genuinely interested in products that you offer.
2. Quantity of traffic:   
   Once you have the right people clicking through from those search engine results pages (SERPs), more traffic is better.
3. Organic results:   
   Ads make up a significant portion of many SERPs.   
   Organic traffic is any traffic that you don't have to pay for.

## **How Search Engine Work?**

search engines are answer machines.   
They exist to discover, understand, and organize the internet's content in order to offer the most relevant results to the questions searchers are asking.

Search engines have three primary functions:

1. Crawl.
2. Index.
3. Rank.

### **Crawl:**

Crawling is the discovery process in which search engines send out a team of robots (known as crawlers or spiders) to find new and updated content.   
Content can vary — it could be a webpage, an image, a video, a PDF, etc. — but regardless of the format, content is discovered by links.

### **Index:**

Store and organize the content found during the crawling process.   
Once a page is in the index, it’s in the running to be displayed as a result to relevant queries.  
Search engines process and store information they find in an index,   
a huge database of all the content they’ve discovered and deem good enough to serve up to searchers.

### **Rank:**

Provide the pieces of content that will best answer a searcher's query, which means that results are ordered by most relevant to least relevant.  
In general, you can assume that the higher a website is ranked, the more relevant the search engine believes that site is to the query.

Reference: <https://moz.com>

## **Google Search Engine:**

Every time you search, there are thousands, sometimes millions, of webpages with helpful information. How Google figures out which results to show starts long before you even type, and is guided by a commitment to you to provide the best information.

Before you search, web crawlers gather information from across hundreds of billions of webpages and organize it in the Search index.

The fundamentals of Search:

The crawling process begins with a list of web addresses from past crawls and sitemaps provided by website owners.   
As google crawlers visit these websites, they use links on those sites to discover other pages. The software pays special attention to new sites, changes to existing sites and dead links.   
Computer programs determine which sites to crawl, how often and how many pages to fetch from each site.  
google offer Search Console to give site owners granular choices about how Google crawls their site: they can provide detailed instructions about how to process pages on their sites, can request a recrawl or can opt out of crawling altogether using a file called “robots.txt”.   
Google never accepts payment to crawl a site more frequently — google provide the same tools to all websites to ensure the best possible results for the users.

Finding information by crawling:

The web is like an ever-growing library with billions of books and no central filing system. google use software known as web crawlers to discover publicly available webpages.   
Crawlers look at webpages and follow links on those pages, much like you would if you were browsing content on the web.   
They go from link to link and bring data about those webpages back to Google’s servers.

Organizing information by indexing:

When crawlers find a webpage, google systems render the content of the page, just as a browser does. google take note of key signals   
— from keywords to website freshness — and we keep track of it all in the Search index.

The Google Search index contains hundreds of billions of webpages and is well over 100,000,000 gigabytes in size. It’s like the index in the back of a book — with an entry for every word seen on every webpage google index. When google index a webpage, google add it to the entries for all of the words it contains.

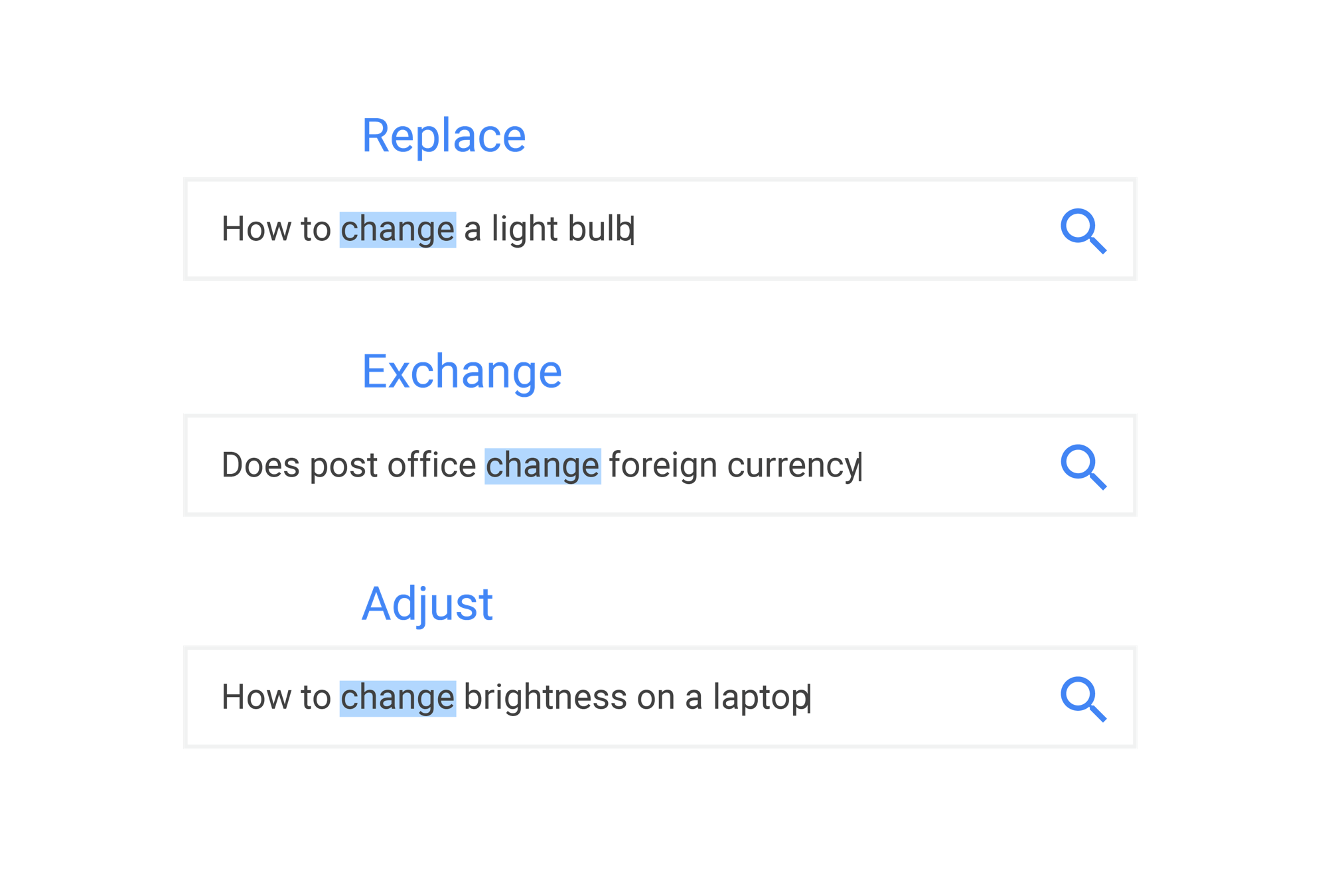
With the Knowledge Graph, we’re continuing to go beyond keyword matching to better understand the people, places and things you care about. To do this, google not only organize information about webpages but other types of information too.   
Today, Google Search can help you search text from millions of books from major libraries, find travel times from your local public transit agency, or help you navigate data from public sources like the World Bank.

**How Search algorithms work:**

With the amount of information available on the web, finding what you need would be nearly impossible without some help sorting through it. Google ranking systems are designed to do just that: sort through hundreds of billions of webpages in Search index to find the most relevant, useful results in a fraction of a second, and present them in a way that helps you find what you’re looking for.

These ranking systems are made up of not one, but a whole series of algorithms. To give you the most useful information, Search algorithms look at many factors, including the words of your query, relevance and usability of pages, expertise of sources, and your location and settings. The weight applied to each factor varies depending on the nature of your query—for example, the freshness of the content plays a bigger role in answering queries about current news topics than it does about dictionary definitions.

the key factors that help determine which results are returned for your query:

1. Meaning of your query  
   To return relevant results for your query, google first need to establish what information you’re looking for and the intent behind your query. Understanding intent is fundamentally about understanding language, and is a critical aspect of Search.   
   google build language models to try to decipher what strings of words should look up in the index.  
   This involves steps as seemingly simple as interpreting spelling mistakes, and extends to trying to understand the type of query you’ve entered by applying some of the latest research on natural language understanding.   
     
   For example, google synonym system helps Search know what you mean by establishing that multiple words mean the same thing. This capability allows Search to match the query   
   “How to change a lightbulb” with pages describing how to replace a lightbulb. This system took over five years to develop and significantly improves results in over 30% of searches across languages.  
     
   Beyond synonyms, Search algorithms also try to understand what category of information you are looking for. Is it a very specific search or a broad query? Are there words such as “review” or “pictures” or “opening hours” that indicate a specific information need behind the search? Is the query written in French, suggesting that you want answers in that language?   
   Or are you searching for a nearby business and want local info?  
   A particularly important dimension of this query categorization is our analysis of whether your query is seeking out fresh content.   
   If you search for trending keywords, algorithms will interpret that as a signal that up-to-date information might be more useful than older pages.
2. Relevance of webpages:

Next, algorithms analyze the content of webpages to assess whether the page contains information that might be relevant to what you are looking for.

The most basic signal that information is relevant is when a webpage contains the same keywords as your search query. If those keywords appear on the page, or if they appear in the headings or body of the text, the information is more likely to be relevant. Beyond simple keyword matching, google use aggregated and anonymized interaction data to assess whether search results are relevant to queries. google transform that data into signals that help machine-learned systems better estimate relevance.

These relevance signals help Search algorithms assess whether a webpage contains an answer to your search query, rather than just repeating the same question. Just think: when you search for “dogs”, you likely don’t want a page with the word “dogs” on it hundreds of times. With that in mind, algorithms assess if a page contains other relevant content beyond the keyword “dogs” — such as pictures of dogs, videos, or even a list of breeds.

It’s important to note that, while our systems do look for these kind of quantifiable signals to assess relevance, they are not designed to analyze subjective concepts such as the viewpoint or political leaning of a page’s content.

1. Quality of content:

Beyond matching the words in your query with relevant documents on the web, Search algorithms also aim to prioritize the most reliable sources available. To do this, google systems are designed to identify signals that can help determine which pages demonstrate expertise, authoritativeness, and trustworthiness on a given topic.

google look for sites that many users seem to value for similar queries. For example, if other prominent websites link to the page (what is known as PageRank), that has proven to be a good sign that the information is well trusted. Aggregated feedback from Search quality evaluation process is used to further refine how systems discern the quality of information.

Spam algorithms play an important role in establishing whether a page is low-quality and help Search ensure that sites don’t rise in search results through deceptive or manipulative behavior. Google’s webmaster guidelines outline the techniques that characterize such low-quality spam sites, including buying links that pass PageRank or sneaking invisible text onto the page.

Content on the web and the broader information ecosystem is constantly changing, and we continuously measure and assess the quality of systems to ensure that we’re achieving the right balance of information relevance and authoritativeness to maintain your trust in the results you see.

1. Usability of webpages:

When ranking results, Google Search also evaluates whether webpages are easy to use. When google identify persistent user pain points, google develop algorithms to promote more usable pages over less usable ones, all other things being equal.

These algorithms analyze signals that indicate whether all users are able to view the result, like whether the site appears correctly in different browsers; whether it is designed for all device types and sizes, including desktops, tablets, and smartphones; and whether the page loading times work well for users with slow Internet connections.

Since website owners can improve the usability of their site, google work hard to inform site owners in advance of significant, actionable changes to Search algorithms. For example, in January 2018 google announced that algorithms would begin to consider the “page speed” of sites, six months before the changes went live. To aid website owners, we provided detailed guidance and tools like PageSpeed Insights and Webpagetest.org so site owners could see what (if anything) they needed to adjust to make their sites more mobile friendly.

1. Context and settings:

Information such as your location, past Search history and Search settings all help us to tailor your results to what is most useful and relevant for you in that moment.

google use your country and location to deliver content relevant for your area. For instance, if you’re in Chicago and you search “football”, Google will most likely show you results about American football and the Chicago Bears first. Whereas if you search “football” in London, Google will rank results about soccer and the Premier League higher.   
Search settings are also an important indicator of which results you’re likely to find useful, such as if you set a preferred language or opted in to SafeSearch (a tool that helps filter out explicit results).

In some instances, we may also personalize your results using information about your recent Search activity. For instance, if you search for “Barcelona” and recently searched for “Barcelona vs Arsenal”, that could be an important clue that you want information about the football club, not the city.

Search also includes some features that personalize results based on the activity in your Google account. For example, if you search for “events near me” Google may tailor some recommendations to event categories google think you may be interested in.   
These systems are designed to match your interests, but they are not designed to infer sensitive characteristics like your race, religion, or political party.

You can control what Search activity is used to improve your Search experience, including adjusting what data is saved to your Google account, at myaccount.google.com.

**Evolving to meet the ever-changing web:**

The web is constantly evolving, with hundreds of new webpages published every second. That’s reflected in the results you see in Google Search: google constantly recrawl the web to index new content. Depending on your query, some results pages change rapidly, while others are more stable. For example, when you’re searching for the latest score of a sports game we have to perform up-to-the-second updates, while results about a historical figure may remain static for years at a time.

Today, Google handles trillions of searches each year. Every day, 15% of the queries process are ones google has never seen before.   
Building Search algorithms that can serve the most useful results for all these queries is a complex challenge that requires ongoing quality testing and investment.

Reference: <https://www.google.com/search/howsearchworks/>

## **Google Search Algorithms Updates:**

In this section we introduce major updates to google search algorithms:

### **Page Rank:**

PageRank (PR) is an algorithm used by Google Search to rank web pages in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages. According to Google:

PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.   
Currently, PageRank is not the only algorithm used by Google to order search results, but it is the first algorithm that was used by the company, and it is the best known.

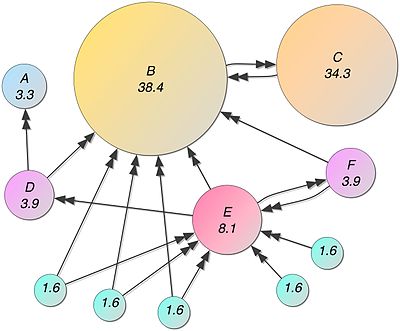
PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references.   
The numerical weight that it assigns to any given element E is referred to as the PageRank of E and denoted by P R ( E ) .   
A PageRank results from a mathematical algorithm based on the webgraph, created by all World Wide Web pages as nodes and hyperlinks as edges, taking into consideration authority hubs such as cnn.com or mayoclinic.org.

The rank value indicates an importance of a particular page. A hyperlink to a page counts as a vote of support. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself.

Numerous academic papers concerning PageRank have been published since Page and Brin's original paper. In practice, the PageRank concept may be vulnerable to manipulation. Research has been conducted into identifying falsely influenced PageRank rankings. The goal is to find an effective means of ignoring links from documents with falsely influenced PageRank.

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process.   
The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

Reference:

<https://en.wikipedia.org/wiki/PageRank>

### **Panda:**

Google Panda Initial Release Date: February 23, 2011

The stated purpose of the Google Panda algorithm update was to reward high-quality websites and diminish the presence of low-quality websites in Google’s organic search engine results. It was also initially known as "Farmer." According to Google, Panda’s initial rollout over the course of several months affected up to 12 percent of English language search results. It get 28 data updates between 2011 and 2015.

**Triggers for Panda:**

The Panda algorithm update addressed a number of problematic phenomena in Google SERPs, including:

* Thin content:   
  Weak pages with very little relevant or substantive text and resources, such as a set of pages describing a variety of health conditions with only a few sentences present on each page.
* Duplicate content:   
  Copied content that appears on the Internet in more than one place. Duplicate content issues can also happen on your own website when you have multiple pages featuring the same text with little or no variation. For example, a chimney sweep company might create 10 pages, one for each city the business serves, with content that is nearly identical on all of the pages with only the city names swapped out (e.g. “We clean chimneys in Denver” on one page and “We clean chimneys in Boulder” on the next, and “We clean chimneys in Aspen” on the next).
* Low-quality content:   
  Pages that provide little value to human readers because they lack in-depth information.
* Lack of authority / trustworthiness:   
  Content produced by sources that are not considered definitive or verified. A Google rep stated that sites aiming to avoid Panda’s impact should work to become recognized as authorities on their topic and entities to which a human user would feel comfortable giving their credit card information.
* Content farming:   
  Large numbers of low-quality pages, often aggregated from other websites. For example, of a content farm might be a website that employs large numbers of writers at a low wage to create short articles covering a vast variety of search engine queries, producing a body of content that lacks authority and value to readers because its core purpose is simply to gain search engine rankings for every conceivable term.
* Low-quality user-generated content (UGC):   
  An example of this type of low-value User Generated Content would be a blog that publishes guest blog posts that are short, full of spelling and grammatical errors and lacking in authoritative information.
* High ad-to-content ratio: Pages made up mostly of paid advertising rather than original content.
* Low-quality content surrounding affiliate links: Poor content around links pointing to paid affiliate programs.
* Websites blocked by users:   
  Sites that human users are either blocking directly in the search engine results or by using a Chrome browser extension to do so, indicating low quality.
* Content mismatching search query:   
  Pages that "promise" to deliver relevant answers if clicked on in the search results, but then fail to do so. For example, a website page might be titled "Coupons for Whole Foods" but when clicked on, there might be no coupons or there might just be a page of ads, leading to disappointment.

**How to recover from Panda:**

* Abandoning content farming practices
* Overhauling website content for quality, usefulness, relevance, trustworthiness and authority
* Revising the ad/content or affiliate/content ratio so that pages are not dominated by ads or affiliate links
* Ensuring that the content of a given page is a relevant match to a user’s query
* Removing or overhauling duplicate content
* Careful vetting and editing of user-generated content and ensuring that it is original, error-free and useful to readers, where applicable
* Using the Robots noindex, nofollow command to block the indexing of duplicate or near-duplicate internal website content or other problematic elements

Reference: <https://moz.com/learn/seo/google-panda>

### **Penguin:**

Google Penguin Initial Release Date: April 24, 2012

Following on the heels of Panda, the Penguin update was announced by Google as a new effort to reward high-quality websites and diminish the search engine results page (SERP) presence of websites that engaged in manipulative link schemes and keyword stuffing.

The initial rollout of Penguin impacted 3.1% of English language search engine queries. Between 2012 and 2016, the filter went through 10 documented updates, evolving over time and influencing the SEO community’s understanding of the problematic practices Penguin sought to address. As of early 2017, Penguin is now part of Google’s core algorithm.

**Triggers for Penguin:**

Penguin targeted two specific practices:

* Link schemes:   
  The development, acquisition or purchase of backlinks from low-quality or unrelated websites, creating an artificial picture of popularity and relevance in an attempt to manipulate Google into bestowing high rankings. For example, an insurance company in Tampa could fill Internet forums with spam comments linking to itself as “best insurance company in Tampa”, falsely inflating its appearance of relevance with these unnatural links. Or, the same company might pay to have links reading “best insurance company in Tampa” appear on an unrelated third-party article about dog grooming; content that has no relationship to the topic.
* Keyword stuffing:   
  Populating a webpage with large numbers of keywords or repetitions of keywords in an attempt to manipulate rank via the appearance of relevance to specific search phrases. For example, an unnatural repetition of keywords on a given page might look like: “AAA Locksmith in Denver, CO is the locksmith in Denver that Denver residents trust when they need a Denver locksmith to quickly help them with their Denver locksmith needs.”

**How to recover from Penguin?**

Unlike a manual link penalty, for which you must file a reconsideration request with Google once you’ve cleaned house, you do not file such a request to have a Penguin penalty lifted. Rather, taking action to remedy problems will often earn ‘forgiveness’ the next time Googlebot comes to crawl your site. These recovery steps include:

* The removal of any unnatural links over which you have control, including links you’ve built yourself or have caused to be placed on 3rd party websites.
* The disavowal of spammy links that you can’t control.
* The revision of your website’s content to remedy over-optimization, ensuring that keywords have been implemented naturally instead of robotically, repetitively or nonsensically on pages where this is no relationship between the topic and the keywords being used

Reference: <https://moz.com/learn/seo/google-penguin>

### **Humming Bird:**

Google Hummingbird Estimated Release Date: August 20, 2013

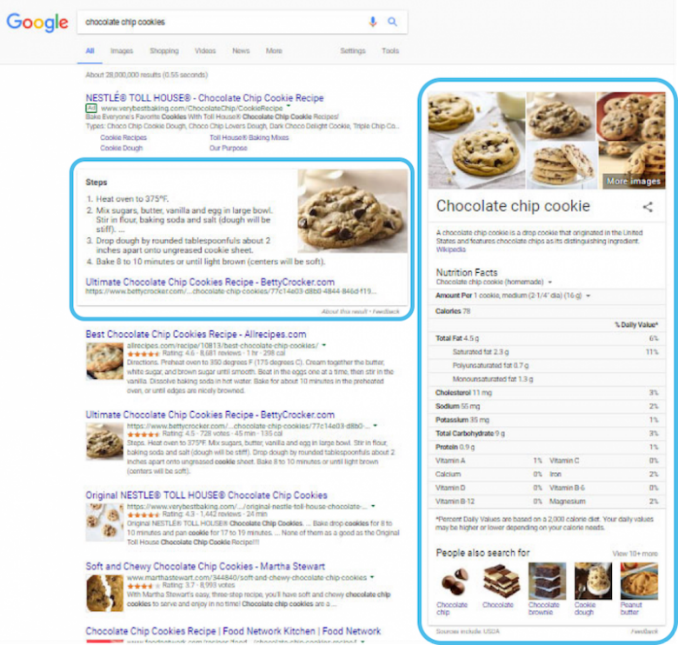
Unlike the previous Panda and Penguin updates which were initially released as add-ons to Google’s existing algorithm, Hummingbird has been cited as a complete overhaul of the core algorithm. While it’s believed that many preexisting components of the core algorithm remained intact, Hummingbird signaled Google’s commitment to an increasingly sophisticated understanding of the intent of searchers’ queries with the goal of matching them to more relevant results.

Google announced Hummingbird on September 26, 2013, but it had actually already been in place about a month prior. Whereas previous algorithm updates like Panda and Penguin sparked significant reporting of lost traffic and rankings, Hummingbird did not appear to have drastic negative impacts on the general web. It was largely understood as having a positive influence on the accuracy of Google’s knowledge base known as the “knowledge graph”. However, the local SEO community theorized that documented effects had been felt in the local search engine results.

**The knowledge graph and semantic search:**

In order to fully understand the purpose behind Hummingbird, it’s important to first be familiar with the search engine features it most heavily impacted: semantic search and the knowledge graph.

A year prior to the release of Hummingbird, Google launched its knowledge graph: not an actual graph, but rather a set of SERP features designed to provide quick, accurate answers to users’ queries about people, places and things.

You can see the knowledge graph in action on the results page yielded upon searching for “chocolate chip cookies”. The SERP does contain standard organic results and links to suitable websites, but it also contains a rich set of knowledge graph data, including an answer box with a recipe, a right-hand knowledge panel featuring nutritional facts about this dessert, and suggestions for related search subjects.

How does Google determine that these results reflect the intent of the searcher and adequately address their needs?   
This is where semantic search comes in.

Semantic search attempts to match appropriate SERP results to the language of Internet users’ queries beyond the meanings of individual keywords, taking a broader context into account, even when the intent of the user is implicit rather than explicit. To see semantic search in action, take a look back at the “chocolate chip cookies” SERP image above. Even though the query was for the cookies themselves, semantic search extrapolated that despite the words used in the search query (which, taken literally, could refer to images or a definition of the cookies), the intent behind the search differed than the language used.

Google then served up results that match the searcher intent rather than searcher language. As a result, we see a SERP full of recipes, nutritional facts, and other topics related to making or eating the cookies.

Google’s increasing mastery of semantic search enables them, in their own words, to understand “real-world entities and their relationships to one another”. Hummingbird’s focus on matching query context to results relies on the intelligence of Google’s technology — the ability to parse intent.

Semantic search is a complex topic, so here’s another example of how this works on the web. Imagine that a Google user types or speaks this search engine query: best place for chinese. It’s only by understanding the intent and context of such a query that Google can identify “place” as signifying a restaurant and “Chinese” as signifying a specific type of restaurant food.

**Hallmarks of Hummingbird:**

One of Hummingbird’s main purposes was to translate semantic search from a concept to a reality — and one that would ultimately become the search standard. It sought to better approximate the true intent of searches so that, for a query like the Chinese food example referenced above, the user receives a set of local Chinese restaurants to choose from, rather than results for the best places for people to live in China or other irrelevant data.

**Hummingbird and voice search:**

In retrospect, the Hummingbird update could be seen as a step on Google’s path toward mastery of the inevitable rise of voice search. When Hummingbird was released in 2013, “conversational search” was making waves in the SEO community. Fast-forward just a few years and the emergence of voice search obliges Google to be equipped to understand fully natural language, using the spoken word for searches like, “Where’s the cheapest place for Mexican food near me?” or “How can I fix a leaky kitchen faucet?”

In sum, technology that is limited to understanding search language keyword-by-keyword isn’t up to the task of gauging intent. Developments like semantic search and an update like Hummingbird are designed to operate effectively in a world of natural language.

**Hummingbird and local search:**

While the organic SEO community strove to understand the effects of Hummingbird on the general web, the local SEO community dealt with some perceived fallout from the update.   
Initially, Hummingbird appeared to populate numerous local results with unsatisfactory “one-boxes”. For example, a query like “denver attorney” would yield a single answer set apart in a box instead of the typical set of local businesses. As many of these one-boxes rewarded businesses that were spamming Google via the use of false business names (e.g. listing themselves as “Denver attorney” instead of their actual name of “Law Office of Jim Davis”), Google was failing to deliver a relevant experience to users. This frustrating scenario was eventually resolved for the most part.

Additionally, the local SEO community speculated that Hummingbird may have had the following effects on the local and local organic results:

* A greater localization of the organic results for queries with a local intent.
* A continued, substantial presence of directory-type results in the local-organic results.

**With Hummingbird, it’s not about recovery — it’s about opportunity!**

Hummingbird signaled to all website owners that Google had its sights set on increased understanding of the true intent behind users’ searches, looking at whole queries for context instead of treating them like strings of keywords.

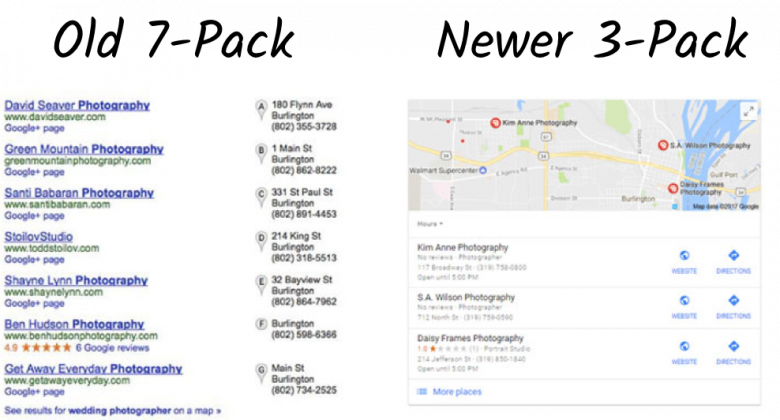
The most practical application of this for website owners is to ensure that natural language is reflected in the site’s content.   
Hummingbird may be seen as bridging the gap between old, spammy practices and modern SEO which seeks to speak the natural language of readers, using their own words.

Reference: <https://moz.com/learn/seo/google-hummingbird>

### **Pigeon:**

GOOGLE PIGEON INITIAL RELEASE DATE: AUGUST 20, 2013

The release of Pigeon resulted in one of the greatest shakeups of Google’s local and local organic results to date. This update was given its moniker by Search Engine Land, an industry publication that received direct intelligence from Google about the intent of Pigeon.

Pigeon was designed to tie Google’s local search algorithm closer to their web algorithm and to improve ranking parameters based on distance and location. The local SEO community reported daily changes to local and local-organic rankings for weeks following the Pigeon rollout, including widespread replacement of the older 7-pack style of local results with packs featuring just 3 listings.

**Hallmarks of Pigeon:**Let’s take a look at some of the major changes that took place as part of the Pigeon update:

* One of Google’s stated purposes for the Pigeon update was to connect their local algorithm more deeply to their traditional web algorithm to take full advantage of the hundreds of ranking signals that go into the web algorithm. These new ties to the web algorithm further emphasized the need for local businesses to have a strong organic web presence in order to compete for local rankings.
* At the same time, Pigeon was slated as featuring improvements to Google’s ability to calculate a local businesses’ distance and location. In many cases, it appeared that the search radius had been narrowed, favoring businesses that were closest to the physical location of the searcher. For example, when searching for a query like “pizza san francisco”, a user in the North Beach area of San Francisco would receive local results narrowed to that neighborhood instead of city-wide results.
* With map boundaries redrawn to a narrower radius post-Pigeon, many business found themselves suddenly outside of the packs in which they were previously ranking. Fast forward several years and Google has become remarkably expert at divining metrics like the distance between a searcher and a business, and Pigeon may be seen most clearly as a step along that path.
* One of the most notable outcomes of Pigeon was the loss of   
  7-pack local results. Over the course of time, Google has steadily decreased the number of local business listings it displays in its local results packs, initially starting with 10 listings and, in 2015, switching to just 3 listings per pack in nearly all cases.

**Pigeon-proofing your web presence:**

Not only the location but there are many factors can be managed for success in a post-Pigeon landscape, including:

* Emphasis on traditional, organic signals: Intelligent link building, publishing exceptional website content and growing Domain Authority all support high local and local organic rankings.
* Emphasis on hyperlocal content: Creating text, video, and image content that associates your business with a specific neighborhood or local region will strengthen your authority as a resource for a narrowed search radius.
* Emphasis on local SEO basics: Competing in the 3-pack environment relies on getting everything right on a basic local SEO checklist. Actions to take include improving the accuracy and breadth of your citations, removing duplicate listings, gaining favorable reviews, and many more fundamental tasks. Large brands should pay particular attention to these foundational   
  to-dos; overlooking proper execution could be holding them back in the local rankings for their hundreds or thousands of locations.
* Spam reporting: When Google fails to police its own local results, you can lend a hand. First, be sure you know Google’s guidelines to the letter, and then take steps to level the playing field for all local businesses by reporting any violations you encounter. Google may not always act on your reports, but it’s worth the effort of trying, especially if demoting a spammer may free up a higher spot for your business. Of course, never report a business just because it’s owned by a competitor. Only report those listings that are violating Google’s guidelines.

### **MobileGeddon:**

What Is Mobilegeddon & The Google Mobile Friendly Update?

**Launch date:** April 21, 2015

**How it works:** Google’s Mobile Update (aka Mobilegeddon) ensures that mobile-friendly pages rank at the top of mobile search, while pages not optimized for mobile are filtered out from the SERPs or seriously down-ranked.

**How to adjust:** Go mobile and focus on speed and usability.  
Google’s mobile-friendly test will help you see which aspects of your page’s mobile version need to be improved.

Reference: <https://searchengineland.com/8-major-google-algorithm-updates-explained-282627>

### **Rank Brain:**

DATE GOOGLE CONFIRMED EXISTENCE OF RANKBRAIN:   
OCTOBER 26TH, 2015

RankBrain is a component of Google’s core algorithm which uses machine learning (the ability of machines to teach themselves from data inputs) to determine the most relevant results to search engine queries.

Pre-RankBrain, Google utilized its basic algorithm to determine which results to show for a given query.

Post-RankBrain, it is believed that the query now goes through an interpretation model that can apply possible factors like the location of the searcher, personalization, and the words of the query to determine the searcher’s true intent. By discerning this true intent, Google can deliver more relevant results.

The machine learning aspect of RankBrain is what sets it apart from other updates. To “teach” the RankBrain algorithm to produce useful search results, Google first “feeds” its data from a variety of sources. The algorithm then takes it from there, calculating and teaching itself over time to match a variety of signals to a variety of results and to order search engine rankings based on these calculations.

**Understanding RankBrain:**

To clearly conceptualize RankBrain, it can help to put yourself in Google’s shoes, trying to understand the intent of a search engine query like “Olympics location.”

What is the true intent of this search? Does the searcher want to know about the Summer or Winter Olympic Games?   
Are they referring to an Olympics that just concluded, or one that will take place four years from now? Is the searcher attending the Olympics right now, sitting in a hotel and looking for directions to the venue for the opening ceremonies? Could they even be looking for historic information about the location of the very first Olympics in ancient Greece?

Now, imagine that in trying to answer this query, all you have is simplistic algorithm signals like the quality of content or the number of links a piece of content has earned to rank results for this searcher. Imagine that the Winter Games in Sochi, Russia just concluded last month and the official Sochi Olympics website has earned millions of links for its content about this past event. If your algorithm is simplistic, it may only show results about the Sochi Games, because they have earned the most links… even if the searcher was actually hoping to learn the location of the next Winter Olympics in Pyeongchang, South Korea.

It’s within this complicated but common situation that the capacity of RankBrain emerges as essential. It’s only by being able to mathematically calculate results based on patterns the machine learning algorithm has “noticed” in searcher behavior that Google can determine that, for example, the majority of people looking up “Olympics location” want to know where the very next Games (be they Summer or Winter) will be held. So, in this case, a Google answer box with the upcoming Games’ location in it will serve the majority of searchers’ needs.

While that answer box may address the intent behind most “Olympics location” searches, there are notable exceptions Google must address.

For instance, if the search is being performed by a user within an Olympic city (like Pyeongchang) the week of the games, Google might instead provide driving directions to the pavilion where the opening ceremonies will be held. In other words, signals like user location and content freshness must be taken into account to interpret intent and deliver the results most likely to satisfy searchers.

RankBrain is a work in progress, with the goal of machine learning perfecting Google’s interpretation of searcher intent over time.

Reference: <https://moz.com/learn/seo/google-rankbrain>

### **Possum:**

GOOGLE POSSUM INITIAL RELEASE DATE: SEPTEMBER 1, 2016

**How it works:**

The Possum update ensured that local results vary more depending on the searcher’s location: the closer you are to a business’s address, the more likely you are to see it among local results. Possum also resulted in greater variety among results ranking for very similar queries, like “dentist denver” and “dentist denver co”. Interestingly, Possum also gave a boost to businesses located outside the physical city area.

**How to adjust:**

Expand your keyword list and do location-specific rank tracking.   
Local businesses now need to be targeting more keywords than they used to, due to the volatility Possum brought into the local SERPs.   
As you check your rankings, make sure you’re doing this from your target location (or, better yet, a bunch of them).

Reference: <https://searchengineland.com/8-major-google-algorithm-updates-explained-282627>

### **Fred:**

**Launch date:** March 8, 2017

**How it works:**

The latest of Google’s confirmed updates, Fred targets websites that violate Google’s webmaster guidelines. The majority of affected sites are blogs with low-quality posts that appear to be created mostly for the purpose of generating ad revenue.

**How to adjust:**

Review Google Search Quality Guidelines and watch out for thin content. If you show ads, make sure the pages they are found on are high-quality and offer relevant, ample information.   
This is basically it: Don’t try to trick Google into thinking your page is about something when it really is a gateway page full of affiliate links. Most publishers make money off ads, and that’s totally legit as long as you are not cheating.

Reference: <https://searchengineland.com/8-major-google-algorithm-updates-explained-282627>

### **Bert:**

Released in October 25, 2019

BERT is not just any algorithmic update, but also a research paper and machine learning natural language processing framework.

**What Is BERT in Search?**

BERT, which stands for Bidirectional Encoder Representations from Transformers, is actually many things.

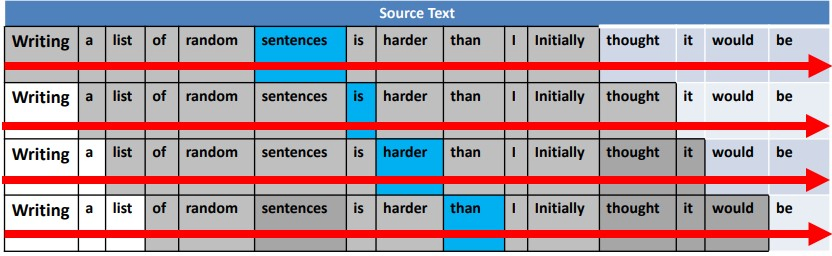
It’s more popularly known as a Google search algorithm ingredient /tool/framework called Google BERT which aims to help Search better understand the nuance and context of words in Searches and better match those queries with helpful results.

**How BERT Works?**

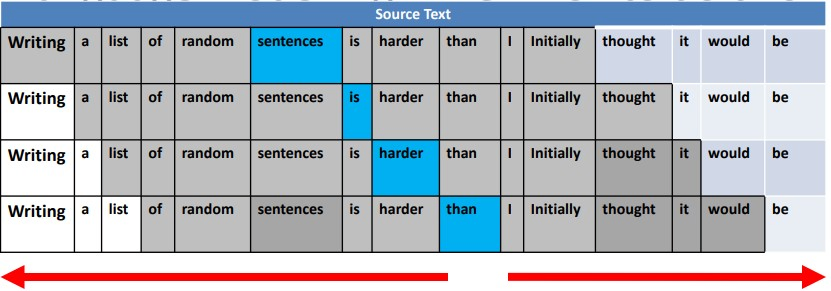
Past language models (such as Word2Vec and Glove2Vec) built context-free word embeddings. BERT, on the other hand, provides “context”.

To better understand how BERT works, let’s look at what the acronym stands for.

**B: Bi-directional:**

Previously all language models were uni-directional so they could only move the context window in one direction – a moving window of “n” words (either left or right of a target word) to understand word’s context. Most language modelers are uni-directional. They can traverse over the word’s context window from only left to right or right to left. Only in one direction, but not both at the same time.

Uni-directional model

BERT is different.   
BERT uses bi-directional language modeling (which is a FIRST).  
BERT can see both the left and the right-hand side of the target word.  
BERT can see the WHOLE sentence on either side of a word contextual language modeling and all of the words almost at once.

Bi-directional model

**ER: Encoder Representations:**

What gets encoded is decoded. It’s an in-and-out mechanism.

**T: Transformers:**

BERT uses “transformers” and “masked language modeling”.  
One of the big issues with natural language understanding in the past has been not being able to understand in what context a word is referring to.  
Pronouns, for instance. It’s very easy to lose track of who’s somebody’s talking about in a conversation. Even humans can struggle to keep track of who somebody’s being referred to in a conversation all the time.  
That’s kind of similar for search engines, but they struggle to keep track of when you say he, they, she, we, it, etc.  
So transformer’s attention part of this actually focuses on the pronouns and all the words’ meanings that go together to try and tie back who’s being spoken to or what is being spoken about in any given context.  
Masked language modeling stops the target word from seeing itself. The mask is needed because it prevents the word that’s under focus from actually seeing itself.  
When the mask is in place, BERT just guesses at what the missing word is. It’s part of the fine-tuning process as well.

**How BERT Will Impact Search?**

BERT Will Help Google to Better Understand Human Language:

BERT’s understanding of the nuances of human language is going to make a massive difference as to how Google interprets queries because people are searching obviously with longer, questioning queries.

BERT Will Help Scale Conversational Search:

BERT will also have a huge impact on voice search.

Expect Big Leaps for International SEO:

BERT has this mono-linguistic to multi-linguistic ability because a lot of patterns in one language do translate into other languages.   
There is a possibility to transfer a lot of the learnings to different languages even though it doesn’t necessarily understand the language itself fully.

**Google Will Better Understand ‘Contextual Nuance’   
& Ambiguous Queries:**

A lot of people have been complaining that their rankings have been impacted. But I think that that’s probably more because Google in some way got better at understanding the nuanced context of queries and the nuanced context of content. So perhaps, Google will be better able to understand contextual nuance and ambiguous queries.

**Should You (or Can You) Optimize Your Content for BERT?**

Probably not.

Google BERT is a framework of better understanding. It doesn’t judge content per search. It just better understands what’s out there.

For instance, Google Bert might suddenly understand more and maybe there are pages out there that are over-optimized that suddenly might be impacted by something else like Panda because Google’s BERT suddenly realized that a particular page wasn’t that relevant for something.

Reference: <https://www.searchenginejournal.com/bert-explained-what-you-need-to-know-about-googles-new-algorithm/337247/>

### **Javascript SEO:**

**What is JavaScript and How is it Used?**

JavaScript is an extremely popular programming language.   
It’s used by developers to make websites interactive.   
JavaScript has the unique ability to **dynamically** update the content of a page.

**Which website elements are commonly generated by JavaScript?**

* Pagination
* Internal links
* Top products
* Reviews
* Comments
* Main content (rarely)

**The complexity of JavaScript crawling:**

1. Googlebot downloads an HTML file.
2. Googlebot finds no links in the source code as they are only injected after executing JavaScript.
3. Googlebot downloads the CSS and JS files.
4. Googlebot has to use the Google Web Rendering Service (a part of the Caffeine Indexer) to parse, compile and execute JavaScript.
5. WRS fetches the data from external APIs, from the database, etc.
6. The indexer can index the content.
7. Google can discover new links and add them to the Googlebot’s crawling queue. In the case of the HTML website, that’s the second step.

**There are multiple reasons why your JavaScript content wasn’t picked up by Google. To name a few:**

* Google encounters timeouts:   
  Are you sure you aren’t forcing Googlebot and users to wait many seconds until they are able to see the content?
* Google had rendering issues: Did you check the URL Inspection tool to see if Google can render it?
* Google decided to skip some resources (i.e. JavaScript files).
* Google decided the content is of low quality.
* It may also happen that Google will index JavaScript content with a delay.
* Google simply wasn’t able to discover this page. Are you sure it’s accessible via the sitemap and the internal structure?

**Different Ways of Presenting JavaScript Content for Google:**

There are several ways of serving your web pages to both users and search engines. And understanding them is crucial when we are talking about SEO, not exclusively in the context of JavaScript. What’s right for your website: Client-side rendering (CSR), Server-side rendering (SSR), or perhaps something more complex?

In the traditional approach (**server-side rendering**), a browser or Googlebot receives an HTML file that completely describes the page. The content copy is already there. Usually, search engines do not have any issues with server-side rendered JavaScript content.

The increasingly popular **client-side rendering** approach is a little different, and search engines sometimes struggle with it. With this approach, it’s pretty common that a browser or Googlebot gets a blank HTML page (with little to no content copy) in the initial load.   
Then the magic happens: JavaScript asynchronously downloads the content copy from the server and updates your screen.

**Best Solution for JS SEO:**

**1. Server-Side Rendering:**

When, for some reason, Google cannot index your JavaScript content, one of the solutions is to implement server-side rendering.   
Websites like Netflix, Marvel, Staples, Nike, Hulu, Expedia, Argos, and Booking.com take advantage of server-side rendering.

There are some tools that can make implementing SSR faster:

|  |  |
| --- | --- |
| Framework | Solution |
| Angular | Angular Universal |
| React | Next.js, Gatsby |
| Vue.js | Nuxt.js |

Quote from Wassim Chegham, a developer expert at Google:   
“One of THE MOST IMPORTANT best practices I’d recommend following is: Never touch the DOM.”

**2. Dynamic rendering**

Another viable solution is called dynamic rendering.

In this approach, you serve users a fully-featured JavaScript website.   
At the same time, your server will send Googlebot (and/or other bots) a static version of your website.

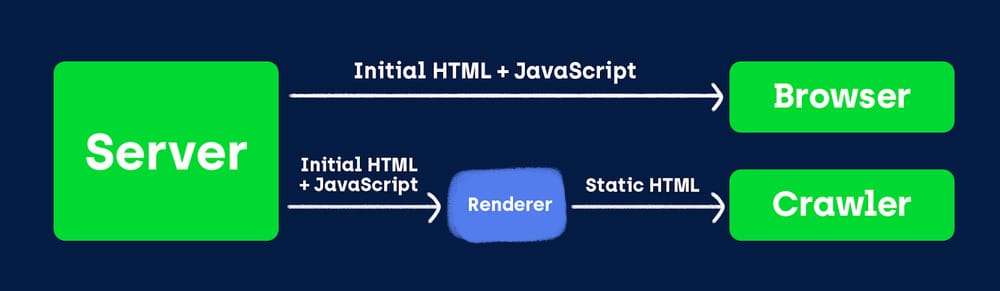
Dynamic rendering is an approach officially supported by Google.

You can use these tools/services to implement dynamic rendering on your website:

Prerender.io

Puppeteer

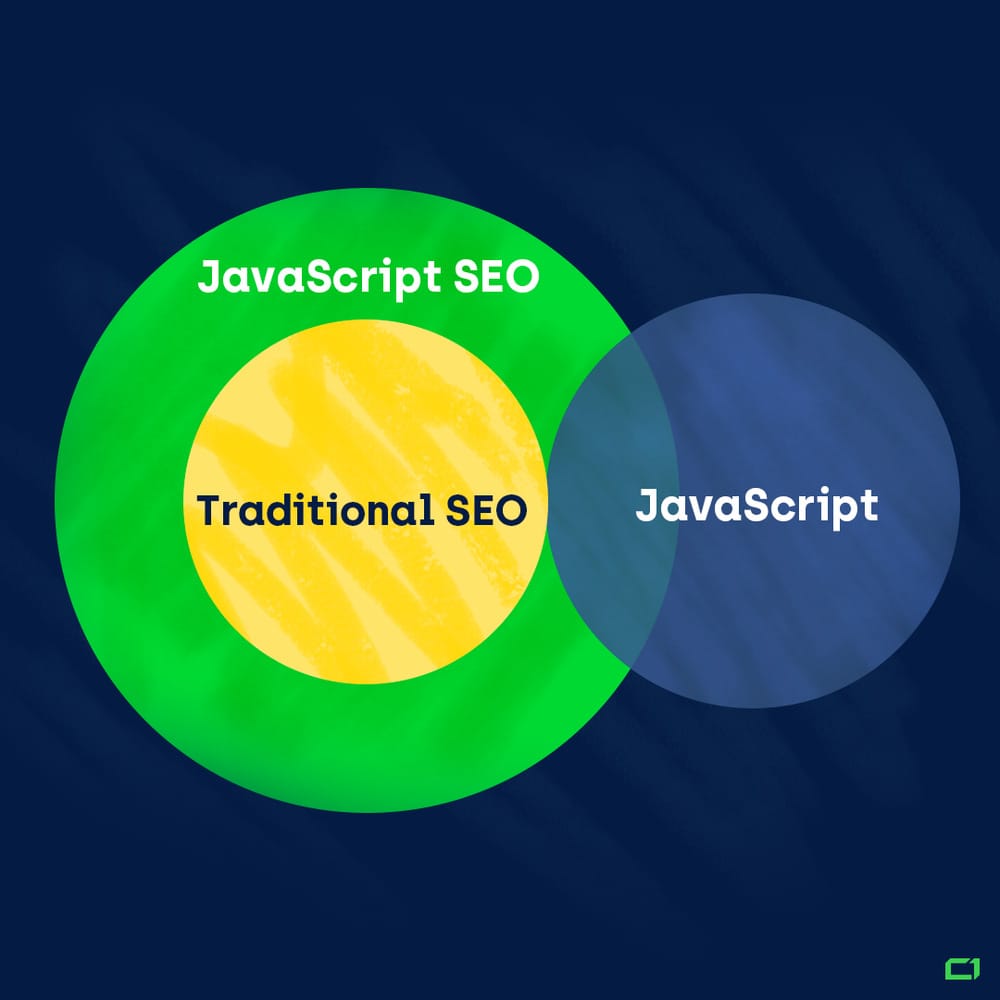
Rendertron



As of 2020, Google recommends using dynamic rendering in two cases:

1. For indexable JS-generated content that changes rapidly.
2. Content that used JS features that aren’t supported by crawlers.

**Takeaways:**

It’s important to remember that JavaScript SEO is done on top of traditional SEO, and it’s impossible to be successful at the former without taking care of the latter.

Sometimes when you encounter an SEO problem, your first instinct might be that it’s related to JavaScript, when in fact, it’s related to traditional SEO.

Apply the principles of SEO well and be very careful before you start blaming JavaScript. Being able to quickly diagnose the source of the problem will save you lots of time.

Here are some other takeaways:

* Google’s rendering service is based on the most recent version of Chrome. However, it has many limitations (may not fetch all the resources, some features are disabled). Google algorithms try to detect if a resource is necessary from a rendering point of view. If not, it probably won’t be fetched by Googlebot.
* Usually, it’s not enough to analyze just the source page (HTML) of your website. Instead, you should analyze the DOM (Right-click -> Inspect tool).
* You can use tools like Onely’s WWJD (What Would JavaScript Do) or browser add-ons to check which elements are generated by JavaScript.
* When you have a JavaScript website and care about traffic from social media, check what your social shares look like (whether they have pictures and custom descriptions).
* You shouldn’t use Google cache to check how Google is indexing your JavaScript content. It only tells you how YOUR browser interpreted the HTML collected by Googlebot. It’s totally unrelated to how Google rendered and indexed your content.
* JavaScript is very error-prone. A single error in your JavaScript code can cause Google to be unable to render your page.
* Use the URL Inspection tool often. It will tell you if Google was able to properly render and index your JavaScript content.
* To make sure Google is able to index your JavaScript you have to take web performance into consideration. Do it for the sake of your users and getting indexed by search engines
* If Google can’t render your page, it can pick up the raw HTML for indexing. This can break your Single-Page Application (SPA), because Google may index a blank page with no actual content.

Reference: <https://www.onely.com/blog/ultimate-guide-javascript-seo/>

## **Modern SEO Mistakes:**

Many old SEO factors became SEO mistakes after google algorithm update in 2019 (Bert update).

So due to SEO experts, modern SEO mistakes are classified in 3 categories:

1. Content writing mistakes.
2. Domain and URLs mistakes.
3. Keywords mistakes.
4. **Content writing mistakes:**

Mistake1:

Regular update your site with fresh new content.

Solution1:

Instead of writing a lot of content, write one article per   
 week or month, and try to promoting this content to get   
 backlinks.

Mistake2:

Short articles.

Solution2:

Google loves long comprehensive content.

Mistake3:

Write for your target audience.

Solution3:

Write for influencers.

1. **Domain and URLs mistakes:**

Mostake1:

Put keyword in the domain.

Solution1:

Keywords in the domain has NO Effect.  
 try to use your brand name.

Mistake2:

Long URLs.

Solution2:

Make your URLs refer to your content either they are long or   
 short.

1. Keyword mistakes:

Before discussion the keyword mistakes, we will explain some terms:

DA: Domain Authority.  
 number of backlinks to our content.

TA: Topic Authority.

LSI: Latent Semantic Indexing

TA and LSI refer to keywords related to specified content.

Search Volume: monthly average searching on a keyword.

So now we will discuss the keyword mistakes:

Mistake1:

Reliance on Domain Authority to rank on google.

Solution1:

Reliance on Topic Authority and LSI to rank.

Mistake2:

Selecting keywords based on search volume.

Solution2:

Don’t reliance on one keyword but include many keywords   
 related to your content.

Mistake3:

Focus on long tail keyword.

Solution3:

Focus on statements and topic.

Mistake4:

Increase keyword density.

Solution4:

Cover all relative keywords in your content.

## **Modern SEO Factors:**

due to SEO experts, modern SEO mistakes are classified in 3 categories:

1. Relevant Content.
2. Backlinks.
3. Google Algorithm.
4. **Relevant Content:**

Google now depend on NLP (Natural Language Processing) to understand the content, so you must introduce quality content to rank on google:

* Long comprehensive content.
* Answers to FAQs.
* Links to high quality related outbound references and websites.
* Media files (videos, infographics, images, GIFs, ……)
* LSI

1. **Backlinks:**

Backlinks are hyperlink to our content from other websites and applications.

Backlinks are double-edged sword because if there are links to your content from irrelated content this will decrease your content’s rank, and also if many not interested searcher open your links and leave directly this will decrease your rank (will explained in next section).

to benefit from backlinks, you should put your links in other related content, or publish them in social media thoughtfully so that only interested searcher will open your links.

1. **Google Algorithm:**

This the basic factor that must be respect to rank on google.

This factor includes many sub-factors to rank:

* Organic CTR from google SERP:  
  CTR: click through rate (the rate of clicking on link in the Search Engine Result Page SERP).  
  to optimize this factor you should optimize <head> tag in the html page by determine the title and meta description due to the content.
* Bounce rate:  
  the time that the searcher spends to move to another page in the same website.  
  to optimize this factor you should decrease the bounce rate by decreasing page loading time so you should decrease page size and media size and you should choose fast hosting like cloud hosting.
* Dwell time:  
  the time that the searcher spends in the page shown in SERP without transfer to another page in SERP.  
  to optimize this factor you should increase Dwell time by improving these elements to ensure that the user stay in your page as long as possible:
* UX (User Experience)
* Internal links
* Related content
* “Thank you” page

# **Project Parts:**

We build our project as 2 applications (GUI & API).

GUI built with Angular and API built with NodeJS & ExpressJS.

Only API have access to the MongoDB database.

Due to MVC design pattern, if the GUI app (View) need to do any operation on database, it will send a request to the API (exactly the Controller in the API), and the Controller will ask the model to execute the operation, then the controller will send the response to the GUI app. Data is exchanged as json.

We partition the project into 3 parts, each part is responsible for many tasks. In this section we will explain each part and its tasks, we will also provide overviews about the third-party libraries used to implement the tasks.

**Project Name:**

Our project name is **“O-Class”** that stands for **“Organized-Class”** because the project is virtual classroom and its objectives are organization and communication.

The project parts are:

* Authentication System
* Blog
* Virtual Classroom Environment

## **Authentication System:**

**Libraries:**

* bcryptjs
* jsonwebtoken
* nodemailer
* randomstring

**bcryptjs:**

we use this library to hash passwords.   
bcryptjs contain function to hash but it doesn’t contain a function to reverse the operation, so we store hashed password in the DB and no one can see the original password.

**jsonwebtoken:**

JSON Web Token (JWT) is an open standard that defines a compact and self-contained way for securely transmitting information between parties as a JSON object. This information can be verified and trusted because it is digitally signed.   
Signed tokens can verify the integrity of the claims contained within it, while encrypted tokens hide those claims from other parties. When tokens are signed using public/private key pairs, the signature also certifies that only the party holding the private key is the one that signed it.  
We use jsonwebtoken library to create the tokens and decrypt the token to ensure that authorized user can do the crucial operations.

**nodemailer:**

this library is used to send email using NodeJS.  
we use it to send welcome message when user sign up and send Reset Password Number when user want to change his password.

**randomstring:**

this library is used to generate random string.  
we use it to generate the Reset Password Number.

**Sign up:**

Teachers and Students can sign up in our website by sending name, email, password and isTeacher or isStudent to the API.  
first the API ensure that the is existed by sending a welcome message to it, the if the email is existed the API hashed the password by bcryptjs and store user data in DB.  
if the user is a teacher, the API will create a folder with the id of the teacher to store the courses files later.

**Log in:**

After sign up the user can log in by sending his email, password and isTeacher or isStudent to the API.  
the API will hash the password then compare sending email and hashed password with the values in the DB. If the user is authenticated the API will generate a token and send it to the user to resend it for each operation need the user to be authorized.

**Log out:**

Delete user data stored in the GUI app or browser loacalStorage or sessionStorage.

**Reset password:**

If you forget your password or want to change your password, you will send the request to the API, then the API will send Reset Password Number generated by randomstring library to your email using nodemailer library, So you enter the Reset Number, and you are transferred to Password Changing page.

**Profile:**

Whether you are teacher or student, you have a profile after sign up.  
The profile contains information about you like name, email and image.  
you can change your information when you want.

## **Blog:**

To implement SEO in our website we build blog to add articles, so google can rank our website and it appears in SERP.

**Libraries:**

We use “@angular/platform-browser” library that is built-in Angular, this library contains 2 classes (Title, Meta).

We use Title class to change the <title> of the page in <head> tag when we transfer to an article page.

We use Meta class to change <meta name=”description” content=””> tag in <head> tag when we transfer to an article page.

By using these classes we make <head> of the page related to <body> of the page, and this will improve the rank of our articles.

**Blog for SEO:**

Our website offers virtual classroom service to teachers and students and google can rank only text content like articles or product description, so we build the blog in our website to make google able to rank our website.

**SEO Factors in the articles:**

We try to implement many SEO factor in our articles like long comprehensive content, LSI, FAQs, images, internal links and mobile friendly in order to access to first page in google.

**Keywords:**

We add these keywords to our articles to cover the LSI, so this increase the chance of google ranking.

oclass, Oclass, OClass, o-class, O-class, O-Class, virtual classroom, virtual classroom websites, classroom, elearning, online class, teacher, student, e-teacher, e-student, eteacher, estudent, elearning platform, elearning environment, virtual environment, virtual classroom environment, course, lesson, homework, quiz

**Google Search Console:**

## **Virtual Classroom Environment:**

**Libraries:**

* socket-io
* multer
* file-manager-js
* random string (mentioned before)

**socket-io:**

socket-io is used for real time communication.  
we use it in discussion of lesson and homework, and also send message between teacher and student.  
we also use it to add announcement in the course.

**multer:**

this library is used to upload files.  
we use it to upload files from GUI app to the API, like changing profile image, changing course image, uploading lesson’s and homework’s files and uploading homework solution.

**file-manager-js:**

this library is used to access to file system and it depends on “fs” module built-in NodeJS.  
we use it to create and delete folders of lessons, homework, courses and teachers.

**Course:**

**Create course:**

Teacher can create course after log in by entering course name, after that the GUI app send request to the API.

The API will:

* generate unique code for the course with randomstring library.
* initiate the course data in DB.
* create folder for the course.

Teacher can delete the course when he wants, so the API will delete course data and folder.

**Update Course:**

After creating course, teacher can update course data like course code, course image, remove students from the course and add lesson, homework, quiz or announcement.

**Enroll course:**

Student can enroll a course by entering the course code which is got from the teacher. The teacher gives course code to students that want to join to his course.

After enrolling course, student can browse the content of the course, download course materials, take homework and quiz and share in discussions.

**Lesson:**

**Add lesson:**

After creating course, teacher can add lesson to the course by entering the name of the course, then the GUI app will send request to the API which initiate lesson data in DB and create folder for the lesson.

There are many materials can be added to lesson:

* video
* files
* links

teacher and student can discuss in the lesson page.

Teacher can delete the lesson when he wants, so the API will delete lesson data and folder.

**Lesson video:**

**Files:**

Teacher can upload files to the lesson, then the students can download them any time.

Uploaded files will be stored in the lesson folder.

Teacher can remove file when he wants.

**Links:**

Teacher can add link to web page as reference to the lesson.

Teacher can also remove the link.

**Discussion:**

Teacher and students can discuss in discussion section on the lesson page.

Messages will send and receive in real time due to socket-io library.

**Homework:**

Homework is very similar to lesson but it contains a section to upload homework solution and there is not video.

When teacher want to add a homework, it will enter homework name and homework date (last time to deliver).

That means, the homework contains files, links and discussion sections, and these sections are the same in the lesson, so we will talk about upload solution section only.

**Upload solution:**

To upload homework solution, student will compress all solution files as one file, then he selects this file to upload.

Student can not upload the solution after homework date ends.

Student can upload one solution and he can’t change the solution after uploading it.

**Quiz:**

**Create a Quiz:**

Teacher can create quizzes in the course by entering quiz name, date & time and questions.

Questions are multi-choices with 4 choices or less.

Teacher determine the true answer and question grades.

Teacher can review questions before creating the quiz.

Teacher can delete the quiz when he wants.

**Quiz Review:**

After creating quiz, teacher can go to quiz-review page that will display questions and grades table.

Grades table will fill automatically when students take the quiz.

**Take a Quiz:**

Students can take quizzes in enrolled courses, he will go quiz page, answer the questions and click finish quiz. He will get his grade immediately.

Student can’t take the quiz if it does not start.

Student can take the quiz only during quiz time.

Student can enter quiz page only one time during quiz time and   
if he refresh the page or go out the page, he can’t complete the quiz.

If student does not finish the during the time, the app will finish the quiz automatically when time is up and student will get his grade.

**Announcement:**

Teacher can add announcement in the course by entering:

* announcement title.
* announcement description.
* From date.
* To date.

Announcement will appear only between “from date” and “to date”.

Teacher can delete the announcement when he wants.

Adding and deleting announcement is in real time.

**Messages Box:**

This feature is similar to formal message sent by email.

Teacher can send messages for his students and student can send message to his teachers.

Messages is sent and received in real time.

Message structure is sender, receiver, subject and content.