

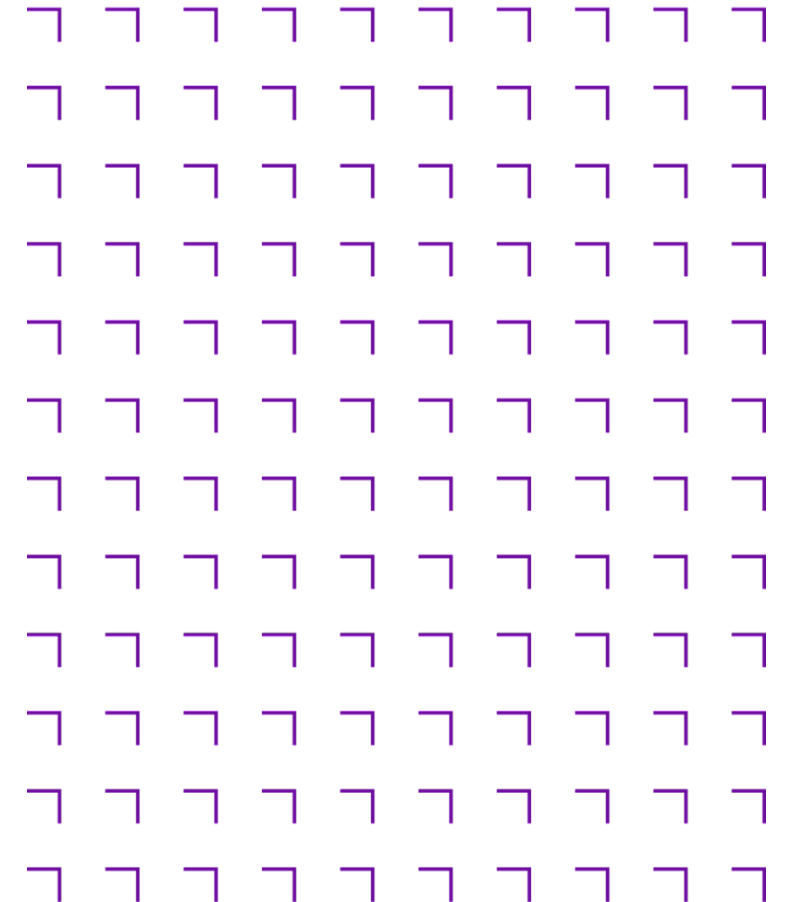
Front-End Web Development

Unit 6: Introduction to websites/web applications

Course Outline



1. Getting Started
2. HTML - Structuring the Web
3. CSS - Styling the Web
4. JavaScript - Dynamic client-side scripting
5. CSS - Making Layouts
- 6. Introduction to Websites/Web Applications**
7. CSS - Advanced
8. JavaScript - Modifying the Document Object Model (DOM)
9. Dynamic HTML
10. Web Forms - Working with user data
11. JavaScript - Advanced
12. Building a Web Application with JavaScript
13. Introduction to CSS Frameworks – Bootstrap
14. Building a Web Application with Svelte
15. SEO, Web security, Performance
16. Walkthrough project



Course Learning Outcomes



- Competently write HTML and CSS code
- Create web page layouts according to requirements using styles
- Add interactivity to a web page with JavaScript
- Access and display third-party data on the web page
- Leverage Bootstrap and Static Site Generator



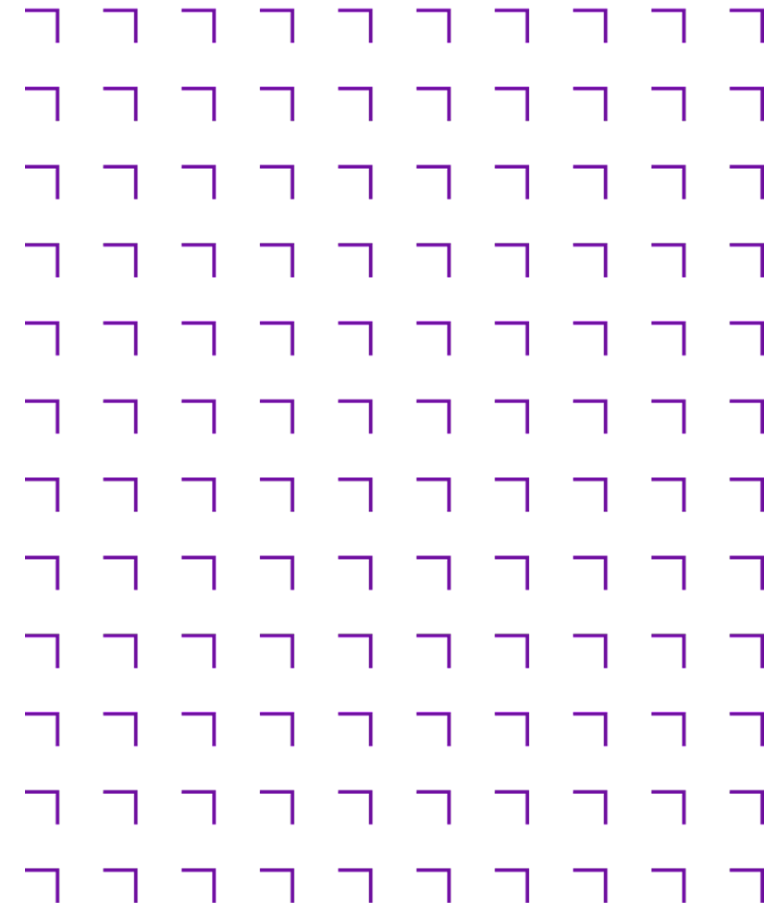
- Final Project - 100% of the grade

- Design and Build functioning Website using HTML5, CSS (including Bootstrap), JavaScript (browser only)

- ✓ Code will be managed in GitHub
 - ✓ Website will be deployed to GitHub Pages
 - ✓ All code to follow best practice and be documented

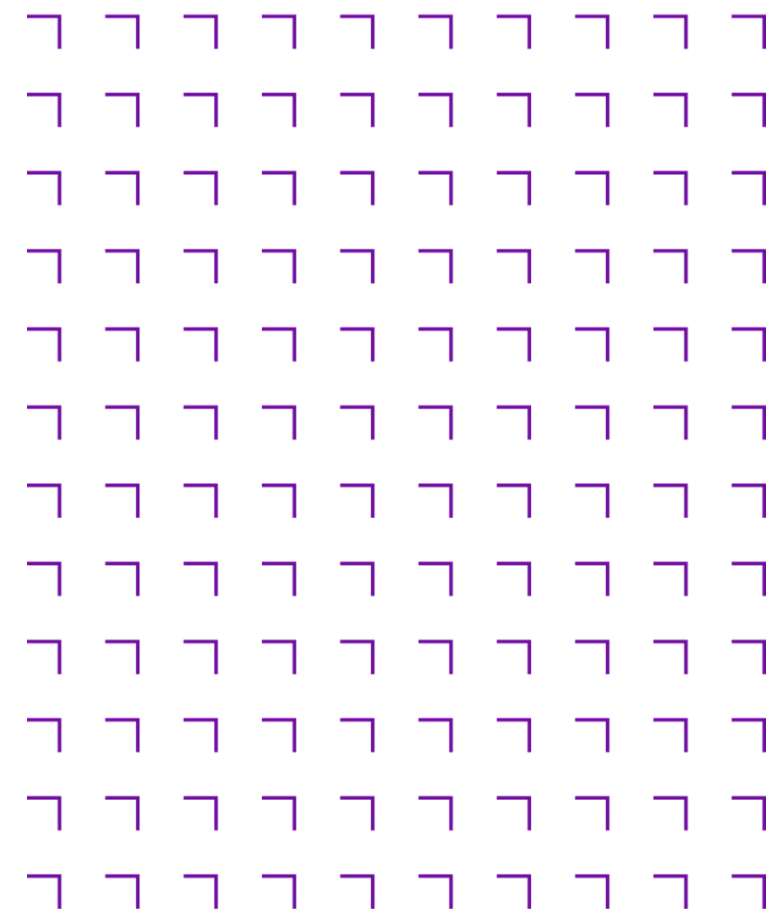
- Details and How-To-Guide are available on the course page under the section called Assessments

Assessment





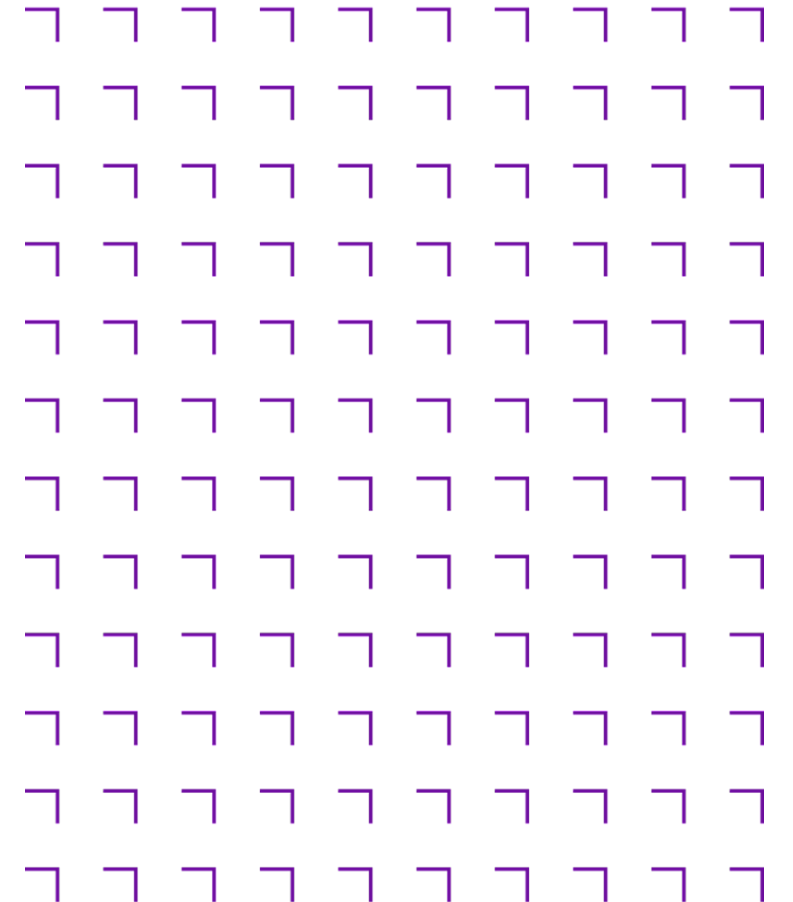
In This Unit



6. Introduction to websites/web applications

Title
Website Development Process
Information Architecture and Site Maps
Create page components, templates/layouts
Build site using GitHub Pages and Astro

Development Process, Information Architecture and Site Maps



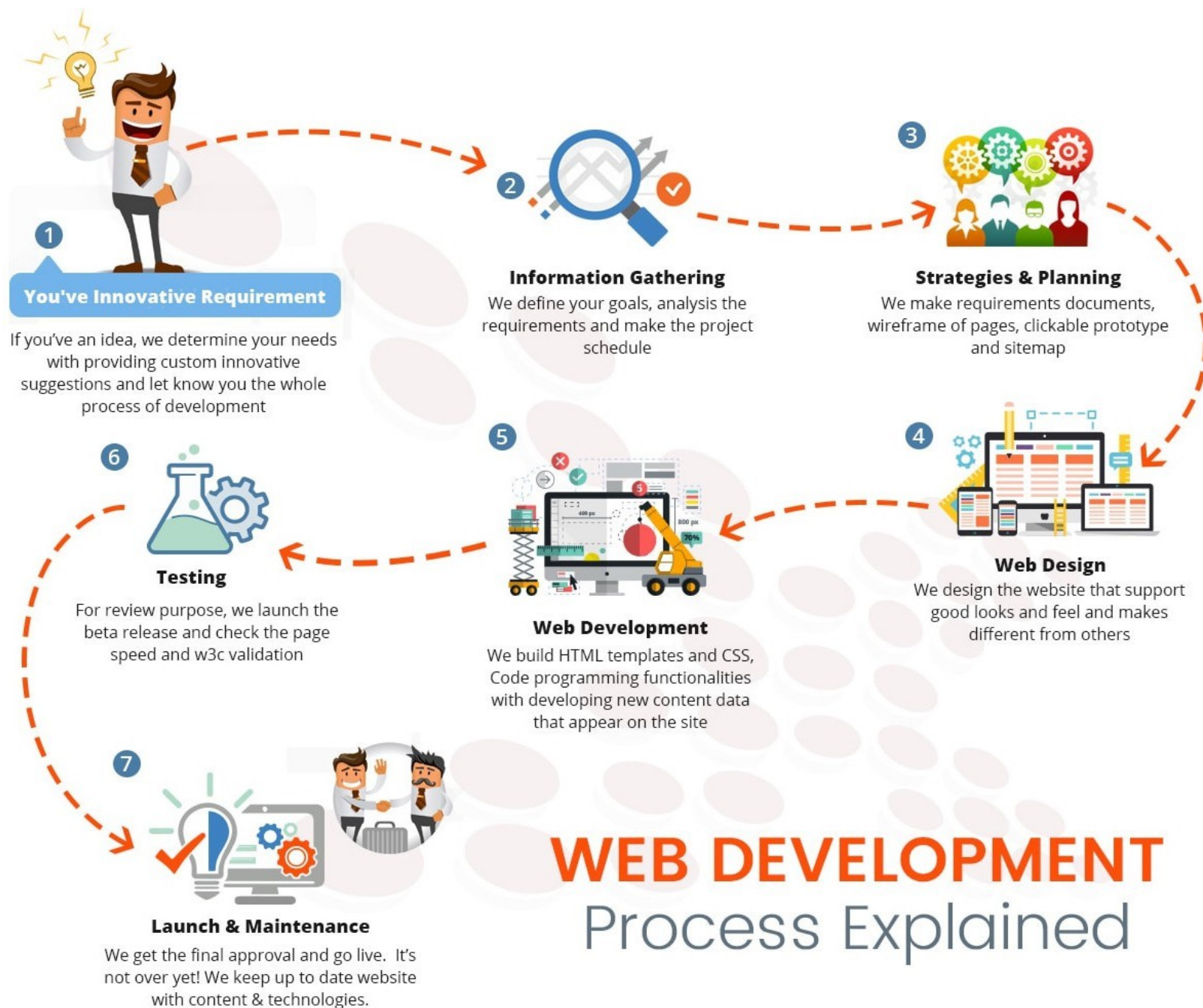
Resources

<https://www.webpixeltechnologies.com/website-development-process-explained/>

<https://prefacestudios.com/insights/our-website-design-process/>

<https://www.smashingmagazine.com/2022/05/top-tasks-focus-what-matters-must-defocus-what-doesnt/>

<https://uxplanet.org/information-architecture-design-a-step-by-step-guide-41dcd4405ee3>



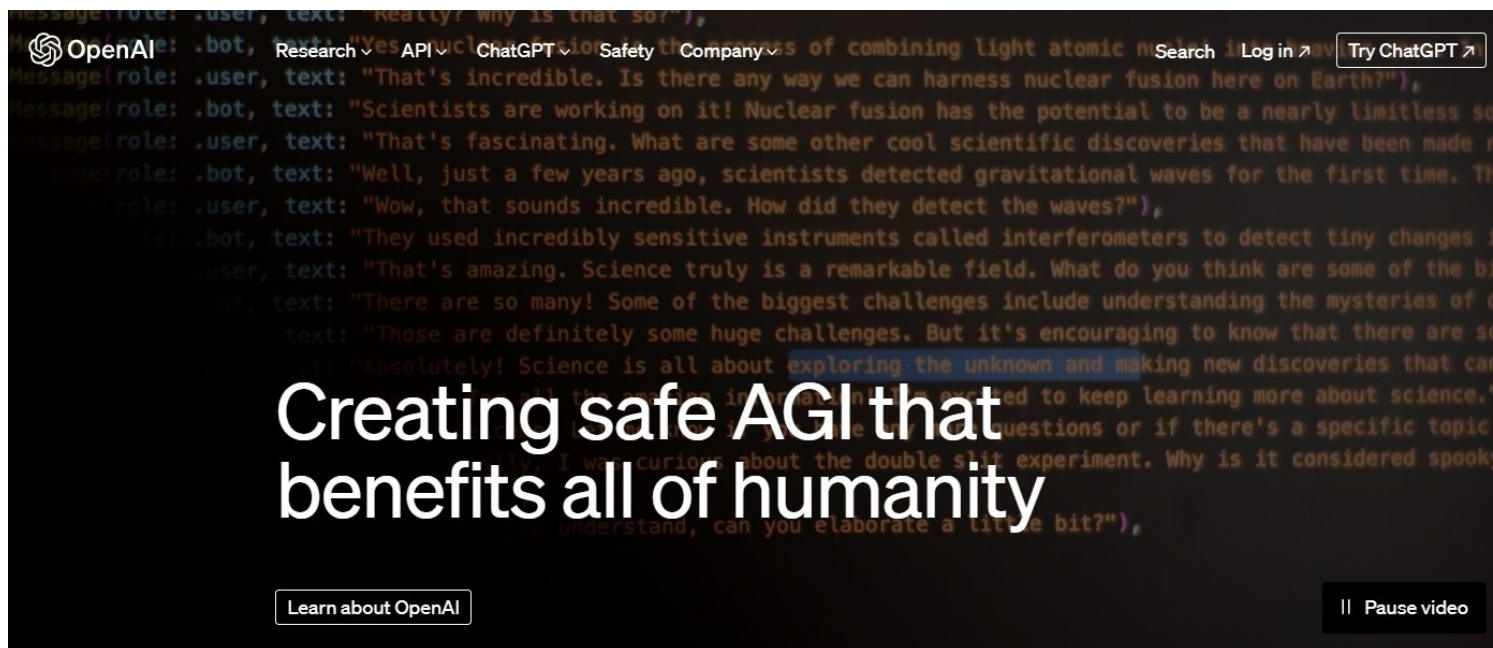
Information Architecture

The big picture of structuring information

"We organize [the information] to understand, to explain, and to control"

— “Information Architecture for the World Wide Web”

The way people understand the world is determined by their ability to organize information



What is Information Architecture (IA)?

Information architecture is the practice of deciding how to arrange the parts of something to be understandable

— [The Information Architecture Institute](#)

- Information Architecture (IA) defines the structure of a website or application, and the relationship between all the areas
- It's the skeleton that teams start working on as early as the first prototypes of the product
- The goal of IA is to classify the content on a website or web application in a way that is understandable, clear, and intuitive
- Then, it needs to be organized in a user-centric way, so that users find what they need effortlessly
- IA combines users and their needs, business goals, and content. As a result, we receive content that is organized and easy to follow

Three circles of Information Architecture



In practice, IA means categorizing content, creating consistent navigation, site map, dividing content into groups

The role of Information Architect

- IA architect/designer is a complement to the design team with the main tasks consisting of research, navigation, data modeling, and labeling
- Information architects often use tools such as card sorting, usability tests, and user interviews
- Information architects gather users needs and content and turn them into something understandable
- They are responsible for how a user will navigate through the web product and reach the relevant information

Information Architecture + UX Design

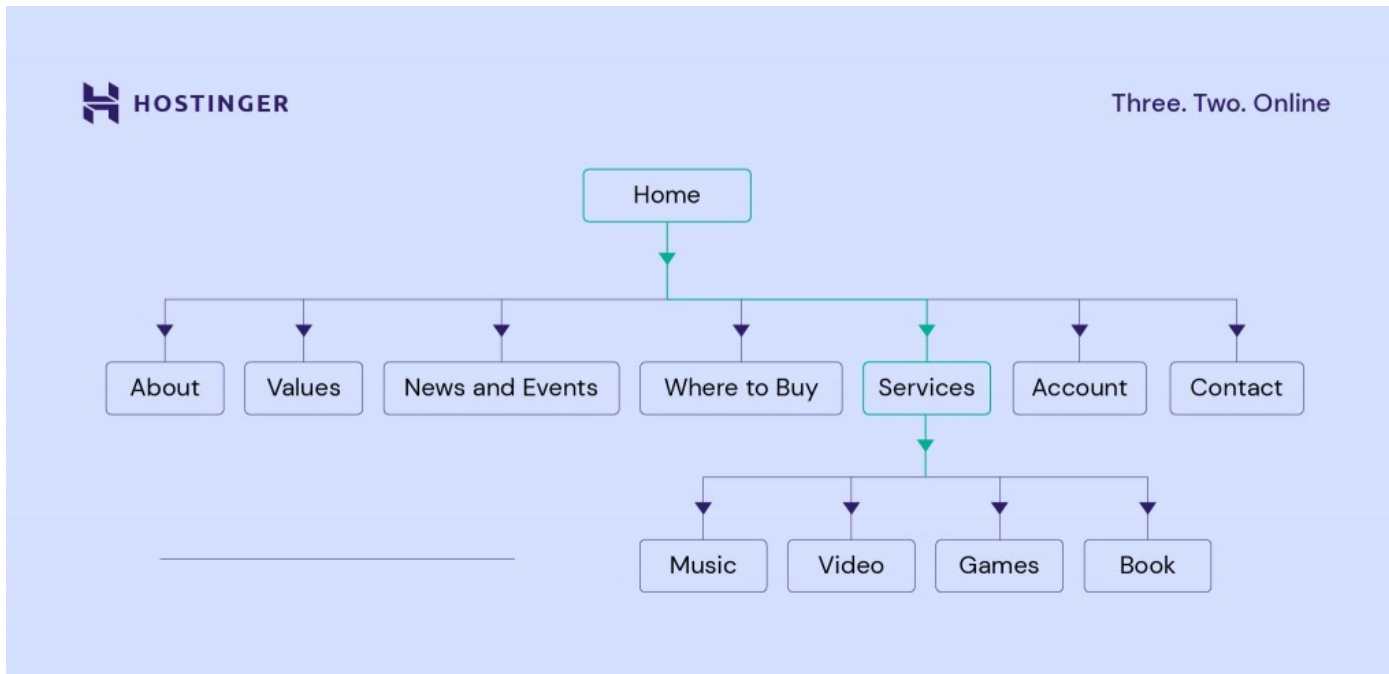
- Information Architecture (IA) is closely tied to the concept of User Experience (UX)
- IA is more about the structure of the entire website or application, while the UX design relates to how a user feels while interacting with them

IA is a blueprint of the design structure which can be generated into wireframes and sitemaps of the project
UX designers use them as the basic materials so that they could plan navigation system

- [UX Planet](#)

Information Architecture vs sitemap

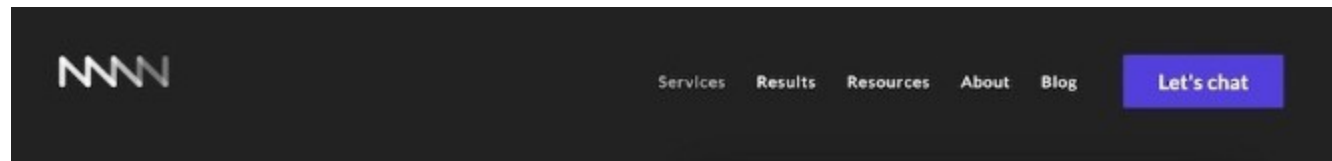
- A sitemap lists all the pages and presents the hierarchy and structure between them
- It also often shows the page goals and functionalities
- Site mapping is a part of IA design
- Sitemaps are based on previously created Information Architecture



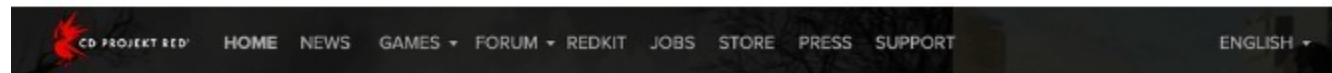
Information Architecture vs navigation

- Navigation guides users through a website or application via links
- Is basically a collection of user interface components like filters, related links, footers, and also global or local on-page navigation
- Navigation is created on the basis of IA, as part of it

Examples of clear global website navigation:



Source: Man.Digital, B2B marketing agency



Source: CD Project Red, gaming industry

Why is Information Architecture important to web design?

The way in which users will use an application or website depends largely on how information is presented and organised

If you've ever tried to use something and thought, "where am I supposed to go next?" or "this doesn't make any sense," you are encountering an issue with an information architecture

— [The Information Architecture Institute](#)

Examples

- You're on a website/app trying to find new runners, and there's no clear navigation or search bar
- Runners are mixed with sandals and boots. How much time will it take to leave the app and go to the competition?
- That's why we need to organize content in a logical way and make it as easy as possible for users to find what they need

Good IA helps us to navigate through tons of information that greets us on every step

Example - [LingsCars.com](https://www.lingscars.com)

We live inside our car leasing website all day Monday to Friday 9am-6pm. We are a broker not a lender. - We are LINGS CARS.com, accept no substitutes

LINGS CARS.com

The UK's craziest car leasing website!
See our 330 reviews on ★ Trustpilot

GET A QUOTE NOW!

WE ARE LINGS YOU CAN TRUST US

Menu

- Home
- Cars / Vans
- Customers
- About Ling
- Fun stuff
- Shop

OUR BEST SELLING CAR LEASING DEALS!

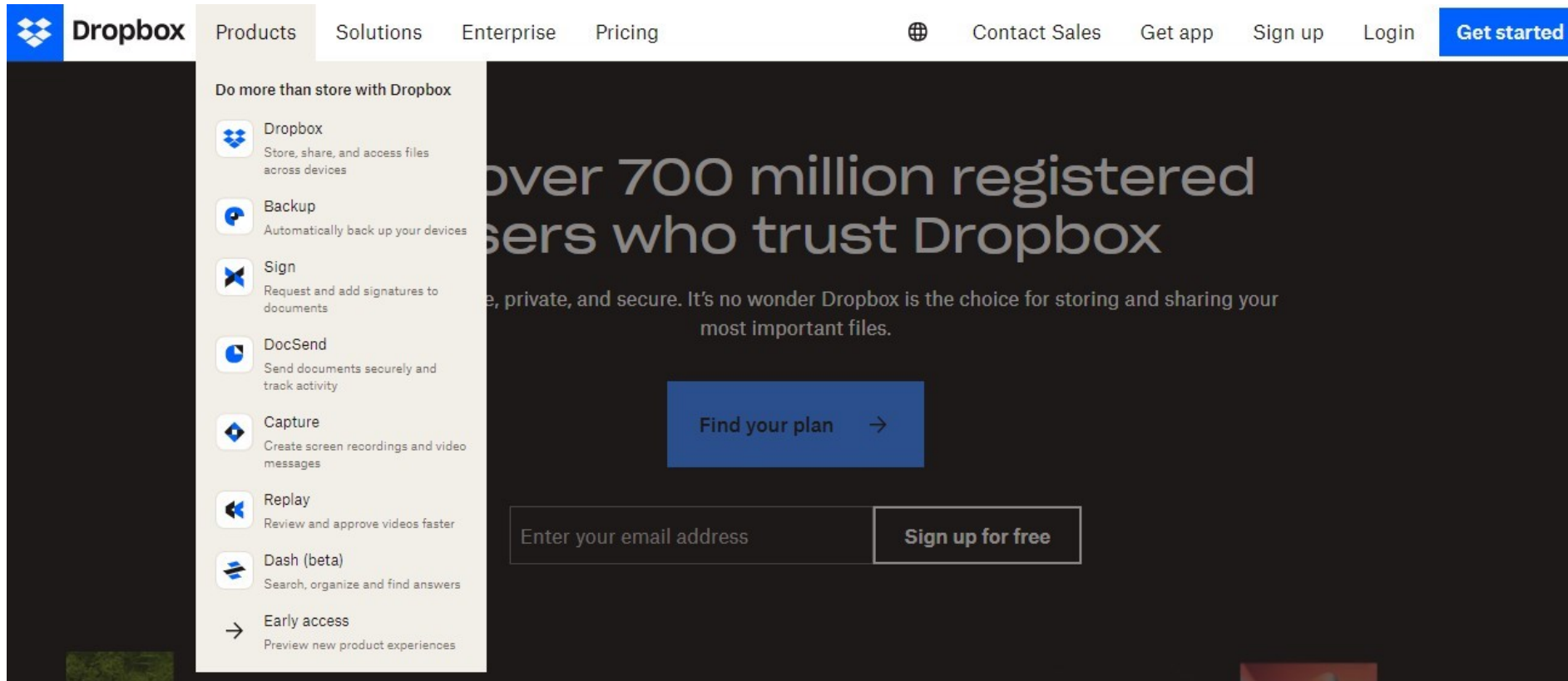
VW T-Cross Estate
1.0 TSI Life (95bhp) Estate 5dr Petrol
3+23

Fiat 500 Electric
87kW La Prima by Borelli 47kWh
3+35

Car Leasing Online Service Response Times
Quote > Proposal > Order > Delivery
80 CUSTOMERS IN 'PROPOSAL'
Ling replies in... **0:17** 51
69 CUSTOMERS IN 'ORDER'

1 DELIVERY LAST WEEK
VW Tiguan Allspace E...
London

Example - Dropbox



Without a good IA, users may be confused after entering your website or application. IA responds to their needs, which oscillate around finding the right information at the right time

The value of Information Architecture for users

- **Reduces information overload**

- Provide clear paths to follow and not too much information on the screen
- Speeds up the process of finding the right information, positive impact on User Experience, reduces frustration

- **Keeps a user focused**

- Minimise distractions for users thus making it easy for them to follow the task they are trying to achieve

- **Helps users to understand where they are**

- Good navigation and Information Architecture help users find the best path to achieve their goal
- The right visual navigation path provides the right context

- **Gives the right information fast**

- Deliver the right content at the right time. Needs to be a fast and efficient process for finding the information

The value of Information Architecture for business

- **Discourages users from escaping to competition**

Good IA with a well-designed navigation and a well-presented value proposition allows you to engage your users better and for a longer time. If users easily find what they need on your website, they most likely won't abandon your website/app and go to the competition from where they will be hard to bring back

- **Increases the chance of conversion and improves ROI**

When your users can find what they're looking for easily and intuitively, with minimum steps possible, the chance of conversion rises. We can achieve that by proper navigation and messaging

- **Reduces the cost of support**

Making it easy to find needed information on your website or application reduces support calls as they will be able to find answers to their questions by themselves

The value of Information Architecture for business

- **Reduces marketing costs**

Poor website structure often results in poor marketing efforts, especially in terms of SEO, Google ranking. If people leave your website or application before completing the desired action, it will be hard to win them back with remarketing because of the bad impression.

- **Improves reputation**

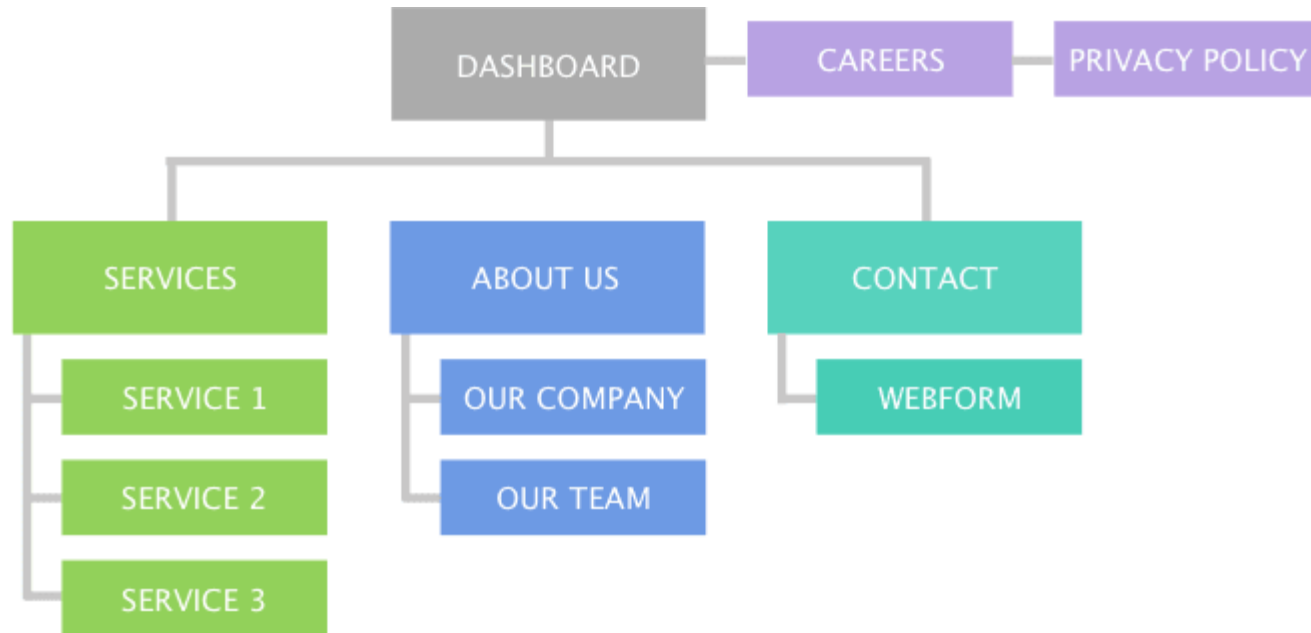
Speaking of an impression, when users can't find what they're looking for and navigate easily on your website or app, it has a negative impact on your brand and its reputation. Good IA and UX design will cause the opposite effect.

Is Information Architecture always needed?

Yes, it is always necessary to think information architecture and navigation through:

- In the case of smaller, simple pages or landing pages it takes less time, but we never should skip that step
- The importance and value of IA will be highly recognized while designing an extensive website or application

Creating a well-designed Information Architecture



Even if you're going to delegate this work to your team or freelancers, **we encourage you to become familiar with the basics of the process**

Information architects seek the answers to the following questions:

- How do you organize and divide information?
- How do you present information?
- How do you communicate the way you navigate through the app?
- How will the user navigate in the application?
- How will the user look for information?

The components that make up a good IA are:

- Organization system
- Labeling system
- Navigation system
- Searching system

A well-designed Information Architecture creates an intuitive and efficient User Experience. For this reason, not only UX skills are necessary to create a good web product, IA skills are equally important. Only then it is possible to create a website or application meeting the users' needs

Tools and templates to help you create an IA

There are a lot of tools for creating Information Architecture. The simplest views can be created using tools such as **Microsoft Visio**

Whiteboard is a visual collaboration plugin for **Adobe XD** that allows to work within the same environment as design files and other workflows

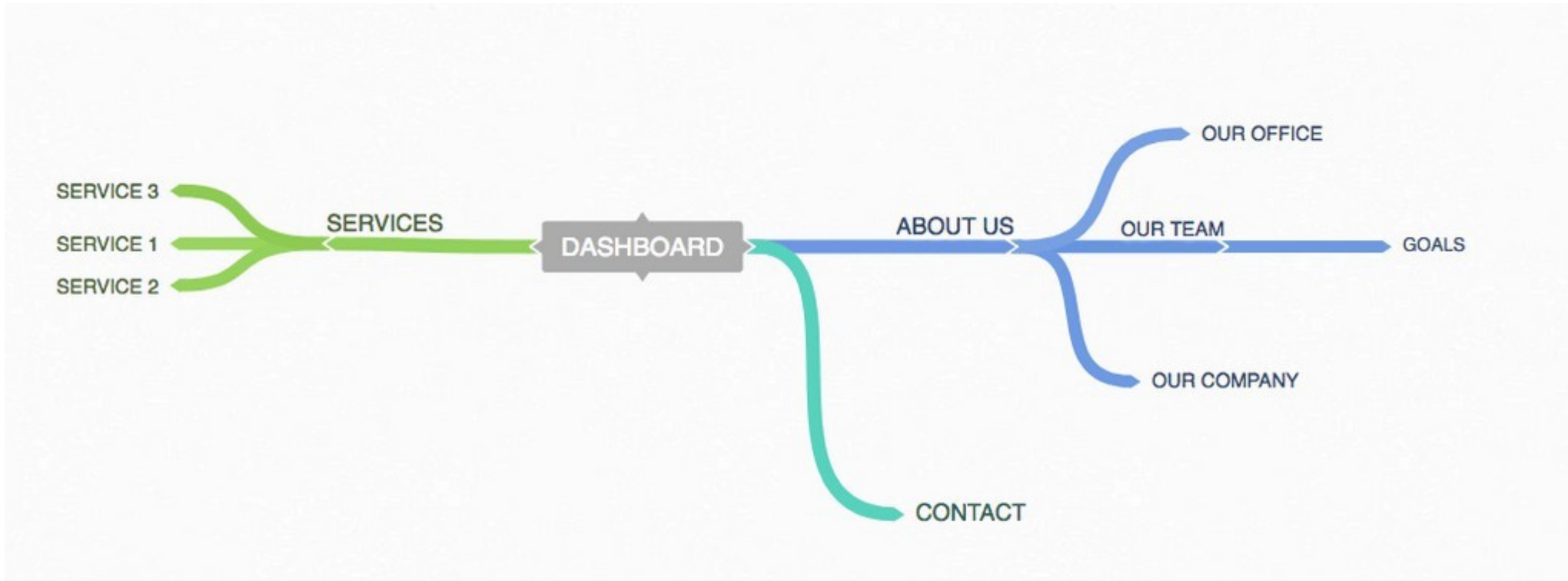
Miro is another visual collaboration tool that supports teamwork with zoomable canvas, web whiteboards, and useful integrations

There are also useful applications for creating diagrams or mind mappings:

- [Create.ly](https://create.ly)
- [OmniGraffle](https://omnigraffle.com)
- [XMind](https://xmind.net)
- draw.io
- [SmartDraw](https://smartdraw.com)
- [Coggle](https://coggle.it)

Presenting Information Architecture

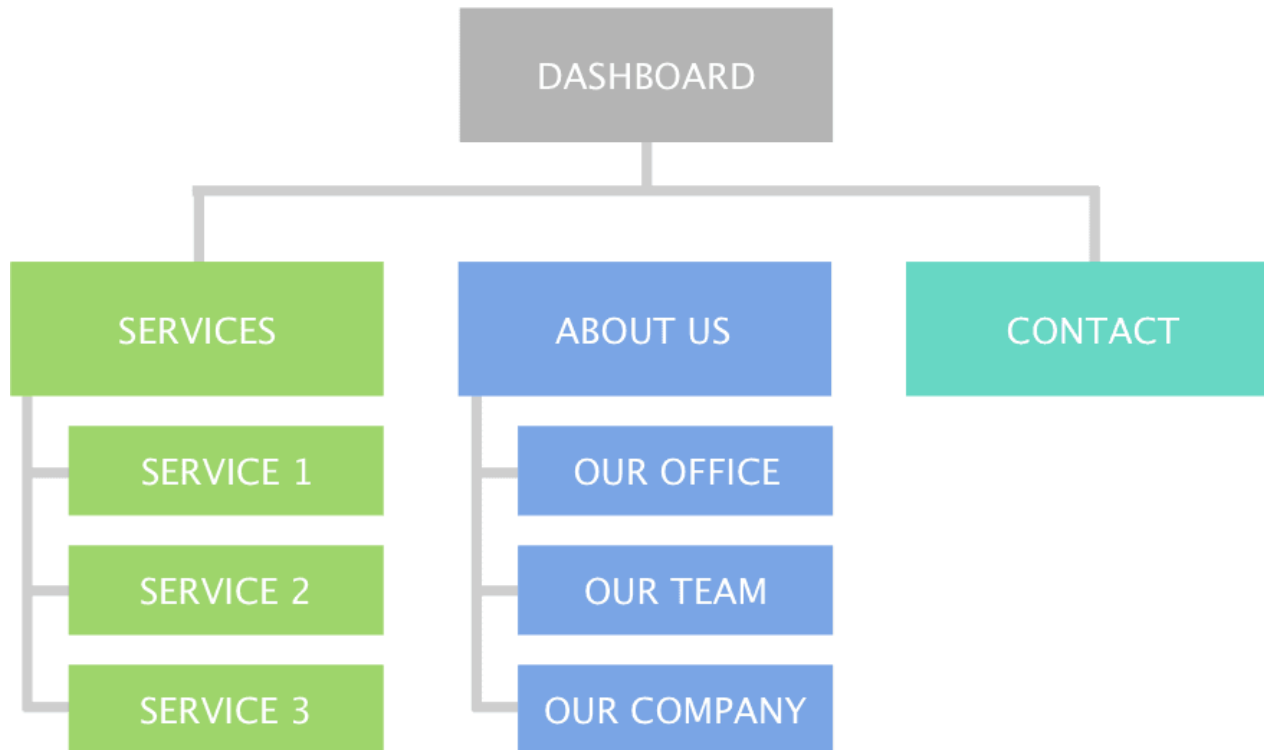
The way of presenting Information Architecture depends on you and your team's preferences. It can be done using mind mapping tools. This way we can effectively visualize the Information Architecture along with its flows



Presenting Information Architecture

Information Architecture can also be presented as a simple diagram and will work for all project sizes

Diagrams are often used for site maps that are based on a previously created IA



How to create Information Architecture for web design?

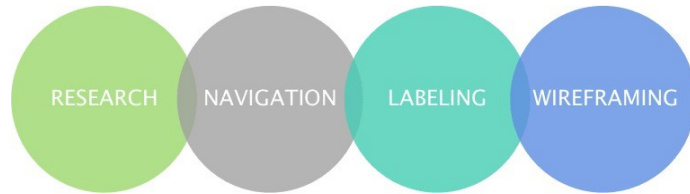
The process of creating IA generally comes up to **4 steps**:

- defining the company's goals
- defining users' goals
- analyzing competitors
- and defining content

If you try to make an existing website or application better, you should also focus on the content audit

To design an effective Information Architecture, your team needs to go through the stages of the design process.

IA Process



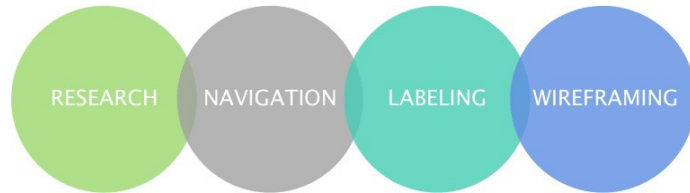
User research

Work on IA should begin with user research and analysis. An important part of this process is to **understand the needs and problems** of product/service users and to reconcile them with the **business goals**. We can use a number of tools for that purpose: card sorting, user interviews, focus groups, or survey. See [Top Tasks](#)

Navigation

The next step is to define the navigation, the **hierarchy of the individual pages**, and to design how and when to display them. At this stage, we can use programs for mind mapping or diagrams

IA Process



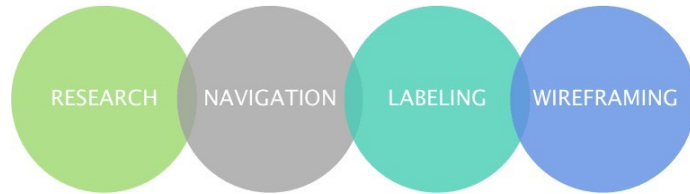
Labeling

Once we've developed the information structure, it's time to **name the sections and pages** so that they are understandable to the user. They also make it easier to navigate the page

Wireframing

The next step is to make wireframes that will be based on IA. By using them, we **define a visual hierarchy of information** on the page as well as connections between subsequent views of the application

IA Process



It is worthwhile to work on Information Architecture at the beginning of the product design process – after recognizing the users of the product, but before creating wireframes and mockups

The first work should include structuring the information on the page, developing navigation, and naming the labels. The next step should be to test the IA on the users, for example using card sorting

IA: the bigger picture of the future

While the tangible targets of our work, websites and software, draw attention and investment, our superpower is organizing words to change minds, and so I propose a new definition: information architecture is the design of language and classification systems to change the world

— Peter Morville, author of the [“Information Architecture for the World Wide Web”](#)

In this quote, Peter Morville refers to the concept of a mental model: a representation of something we — as users, and as people — have in our minds. They are built in our brains based on what we’ve learned and experienced while interacting with products and the world

By improving our mental models — we’re learning. And by improving mental models of others — we’re teaching. “Improving others’ mental models means we’re teaching, and it’s in this uncharted stretch of IA where treasure lies.” — says Morville

Information Architecture from a business perspective is one thing. But we never should underestimate its other power: organizing to teach, shape minds, and change the world

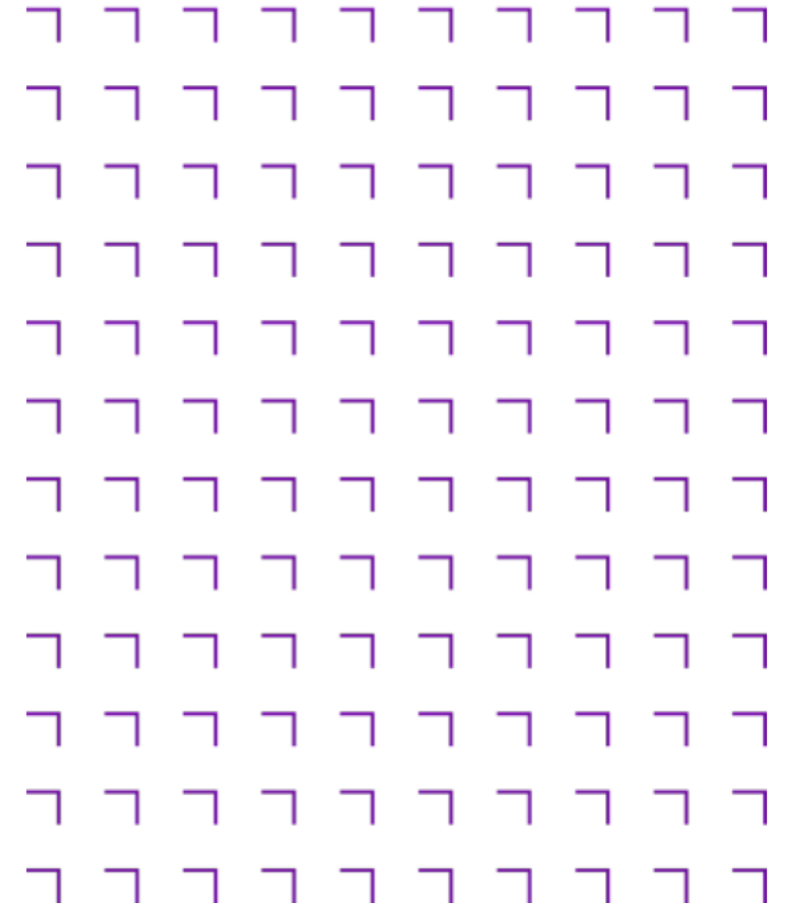
Next Steps: Learn more about [User Experience design in Software Development](#)



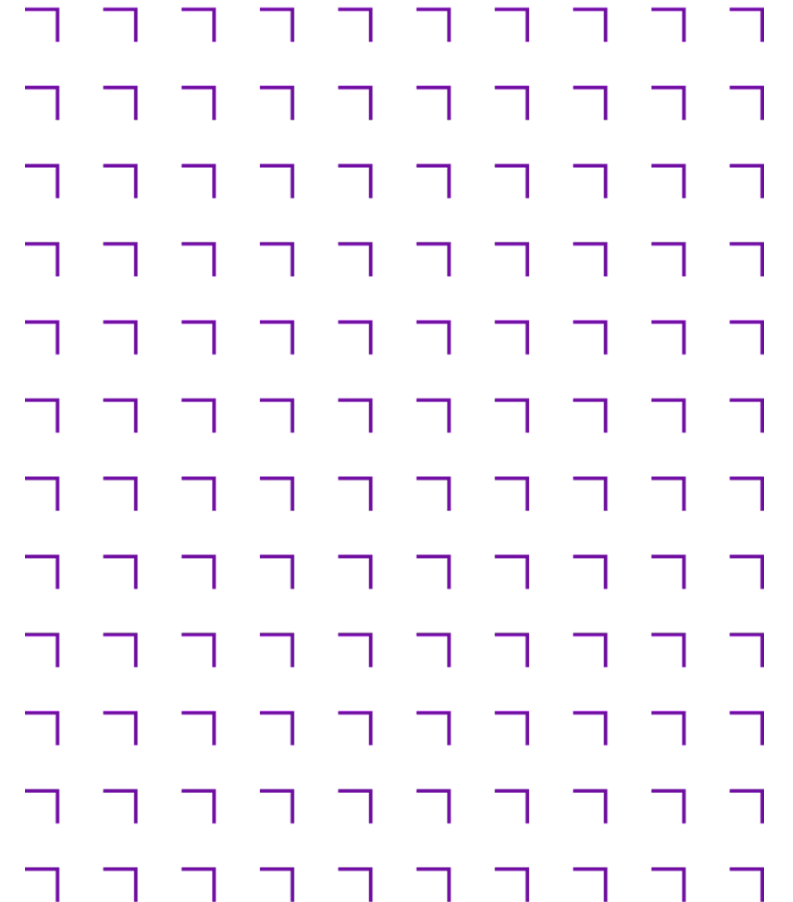
Activity

Breakout

- Join a breakout room
- Download the unit 6 exercises code from Moodle
- Follow the instructions and complete the exercises
- You have 35 minutes
- Lecturer will visit each room in turn, etc...
- Will start next topic on the hour



Create page components, templates/layouts



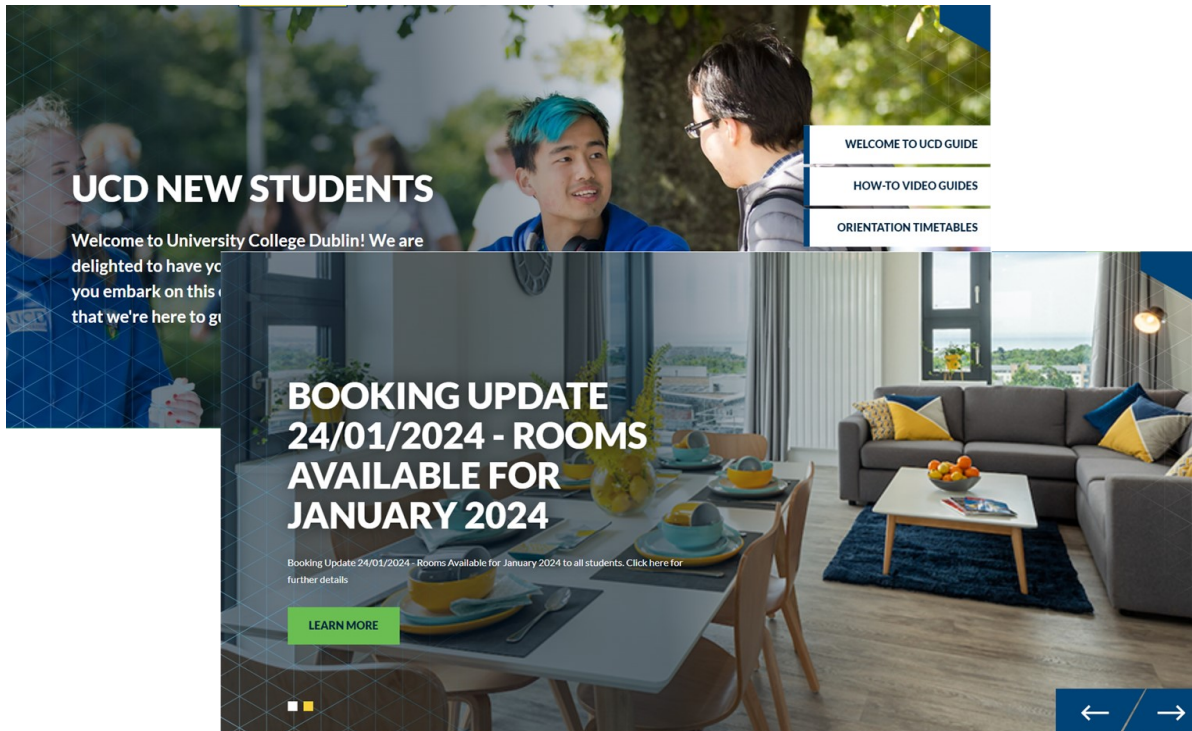
Resources

- <https://simplicate.ca/blog/component-based-website-design>
- <https://www.gorrion.io/blog/what-is-a-design-system-and-do-you-really-need-one/>
- <https://atomicdesign.bradfrost.com/>
- <https://bradfrost.com/blog/post/extending-atomic-design/>

What Is Component-Based Web Design

It's an approach to web design that starts by building the types of components that suit your content with the intent of creating a library of reusable content blocks

Each component has a specific purpose and is designed so that it can be placed in different locations within a page but also on different pages and with different content. See [UCD.ie](https://www.ucd.ie)



Page Based Design

Traditionally organizations built their websites with a page based design in mind

With a page based design, you think about the page you need first, set that up, and then build components within it to meet different specifications. The process works something like this:

- Designers mock-up a page and get it approved by stakeholders
- A Front End Developer builds out a page to match the mock-up
- A Back End Developer writes custom components to the page
- The team responsible for content gets trained which components go to which pages
- Repeat this over and over again for every single page

While teams working on these kinds of projects can find ways to streamline the process, they're inevitably left with a repository of content that grows out of hand quickly

As a company's brand grows and team members come and go, components built for a page based design become obsolete and get buried under a mountain of new pages. In short, a page based design simply can't scale in size without causing a lot of pain

Component Based Design

It's an approach to web design that starts by building the types of components that suit your content with the intent of creating a library of reusable content blocks that don't go out of style

In short, it's an approach that flips traditional web design on its head, designing websites from the bottom-up, rather than the top-down

Components are composed of HTML (Markup and content), CSS and JavaScript.

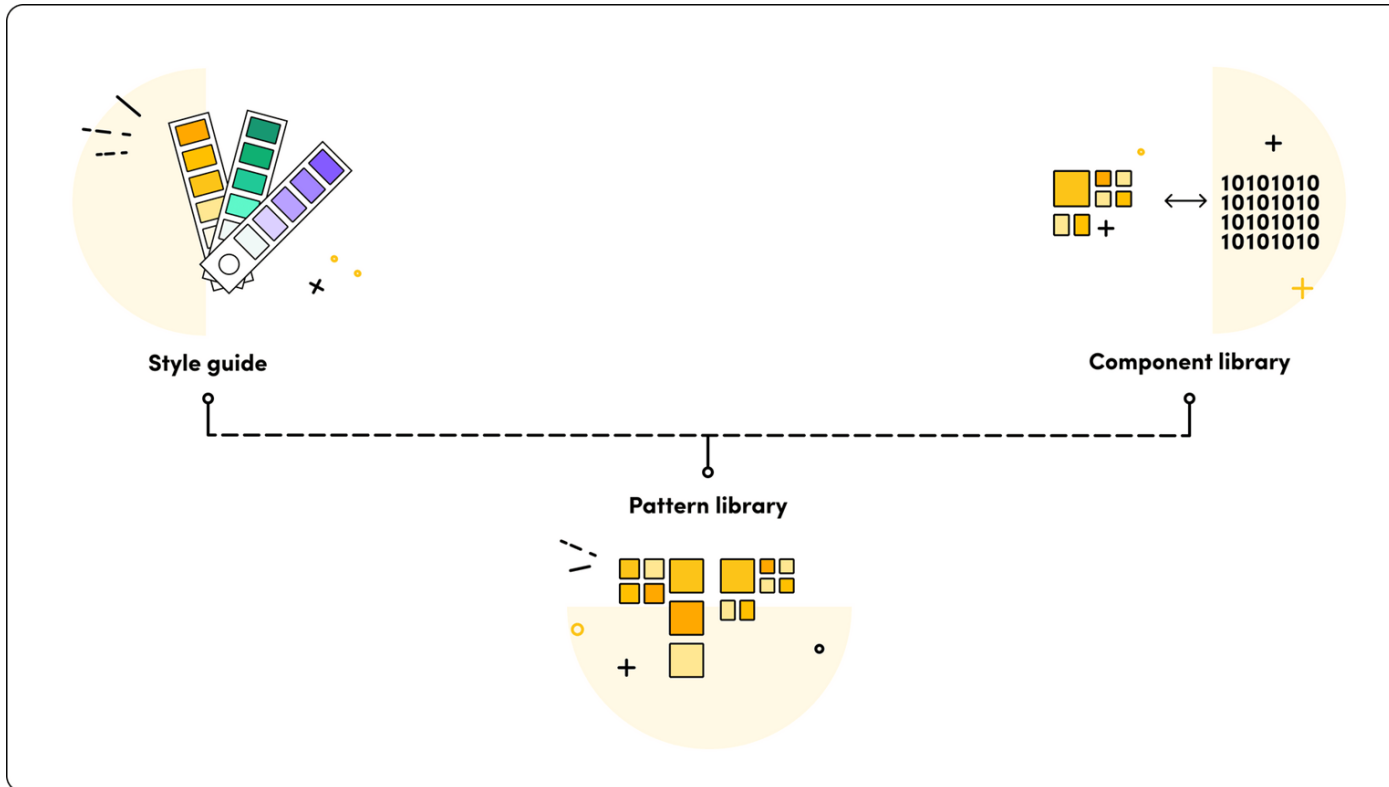
Component-based web development offers several benefits:

- fast and flexible development of the web
- application code reuse and modularity
- enhanced code readability and maintainability
- improved user interface design and performance

Examples include: [Bootstrap](#), [Tailwind UI](#), [Foundation Blocks](#), [Material Design](#)

Design Systems

Can be defined as a set of reusable UI elements guided by standards and requirements, thanks to which specific organizations started to create their visual identity. It usually consists of several parts – components, brand guidelines, code standards, style guides, tokens and UI Kit



Atomic Design: A Methodology for Building User Interfaces

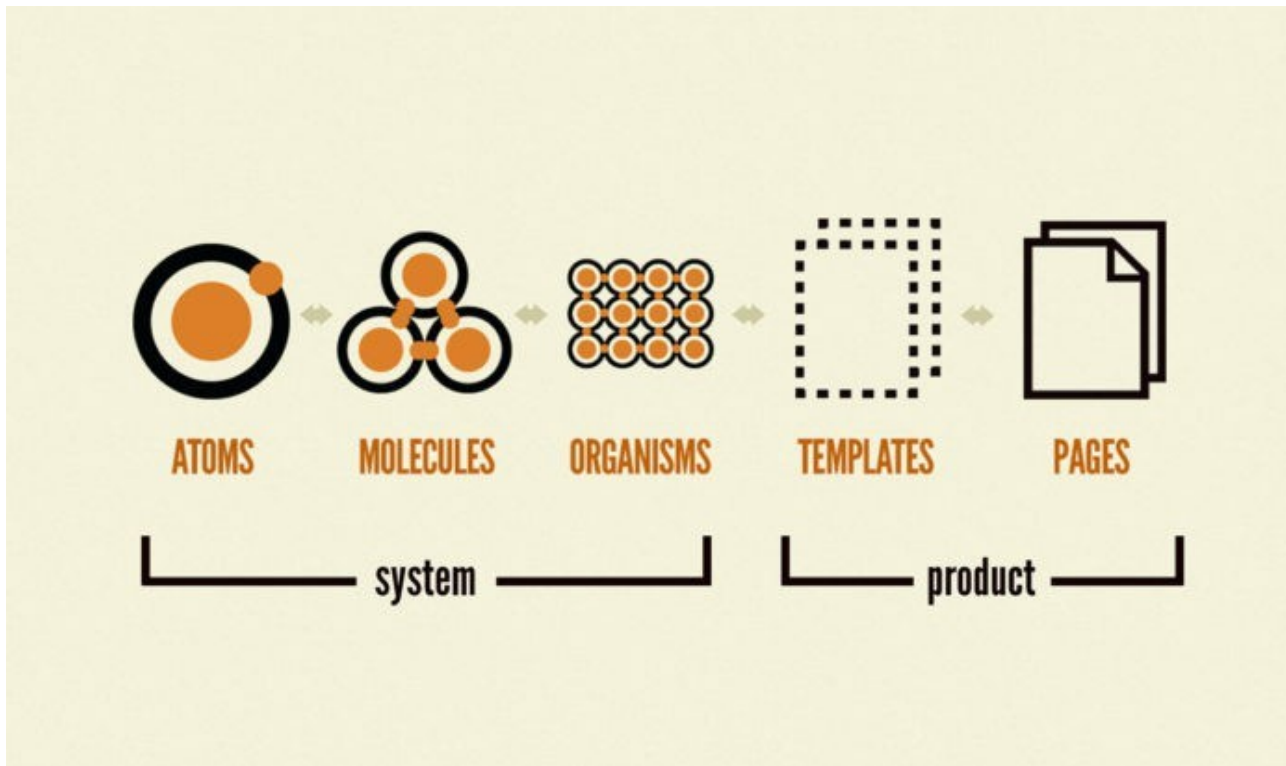
A Practical Approach to Creating Consistent, Reusable, and Scalable Design Systems

Create design systems, not pages

Examples: [Salesforce Lightning Design System](#), [Shopify Polaris](#), [Google Material Design](#)

Atomic Design - Introduction

- [Atomic Design](#) is a methodology for creating user interfaces and digital products
- It breaks down a design system into smaller, more manageable components
- These components can be combined to create more complex components



Atomic Design - The Five Levels of Atomic Design

Atoms:

The most basic building blocks of a design system, example: Typography, color, buttons, icons

Molecules:

Groups of atoms that work together to form a more complex element, example: A form field, a navigation menu

Organisms:

Groups of molecules that work together to form a larger component, example: A header, a footer

Templates:

Complete pages or layouts that are made up of organisms and molecules, example: A homepage, a product page

Pages:

Instances of templates that are filled with content, example: A specific page with product details, a shopping cart page

Atomic Design - Benefits

Consistency:

Promotes consistency by ensuring that all components use the same styles and patterns

Reusability:

Can be reused across different projects and pages, saving time and effort

Scalability:

Systems can be easily adapted to new devices and platforms

Atomic Design - When to Use it

- **Building a Scalable Design System:**

Is a great way to create a reusable library of design elements that can be scaled across different projects and platforms

- **Collaborating with Designers and Developers:**

Provides a common language and framework for designers and developers to work together effectively

- **Creating a Consistent Brand Identity:**

Can help create a consistent visual language that reinforces the brand identity across all touchpoints

- **Designing for Multiple Devices and Platforms:**

Components can be adapted to different screen sizes and resolutions, ensuring a consistent user experience across devices

- **Iterating and Refining Designs:**

Makes it easy to make changes to components without affecting the overall design system

Atomic Design - Applying it

- **Define your atoms:**

Identify the basic building blocks of your design system, such as typography, colors, and buttons

- **Create your molecules:**

Combine atoms to form more complex elements, such as form fields, navigation menus, and buttons

- **Build your organisms:**

Combine molecules to create larger components, such as headers, footers, and product pages

- **Create your templates:**

Build complete pages or layouts using organisms and molecules

- **Implement your pages:**

Fill templates with content to create final pages

Atomic Design - Tips for Success

- **Use a design system:**

Create a centralized repository for your design elements and patterns

- **Collaborate with your team:**

Regularly communicate and collaborate with designers, developers, and stakeholders

- **Test your design system:**

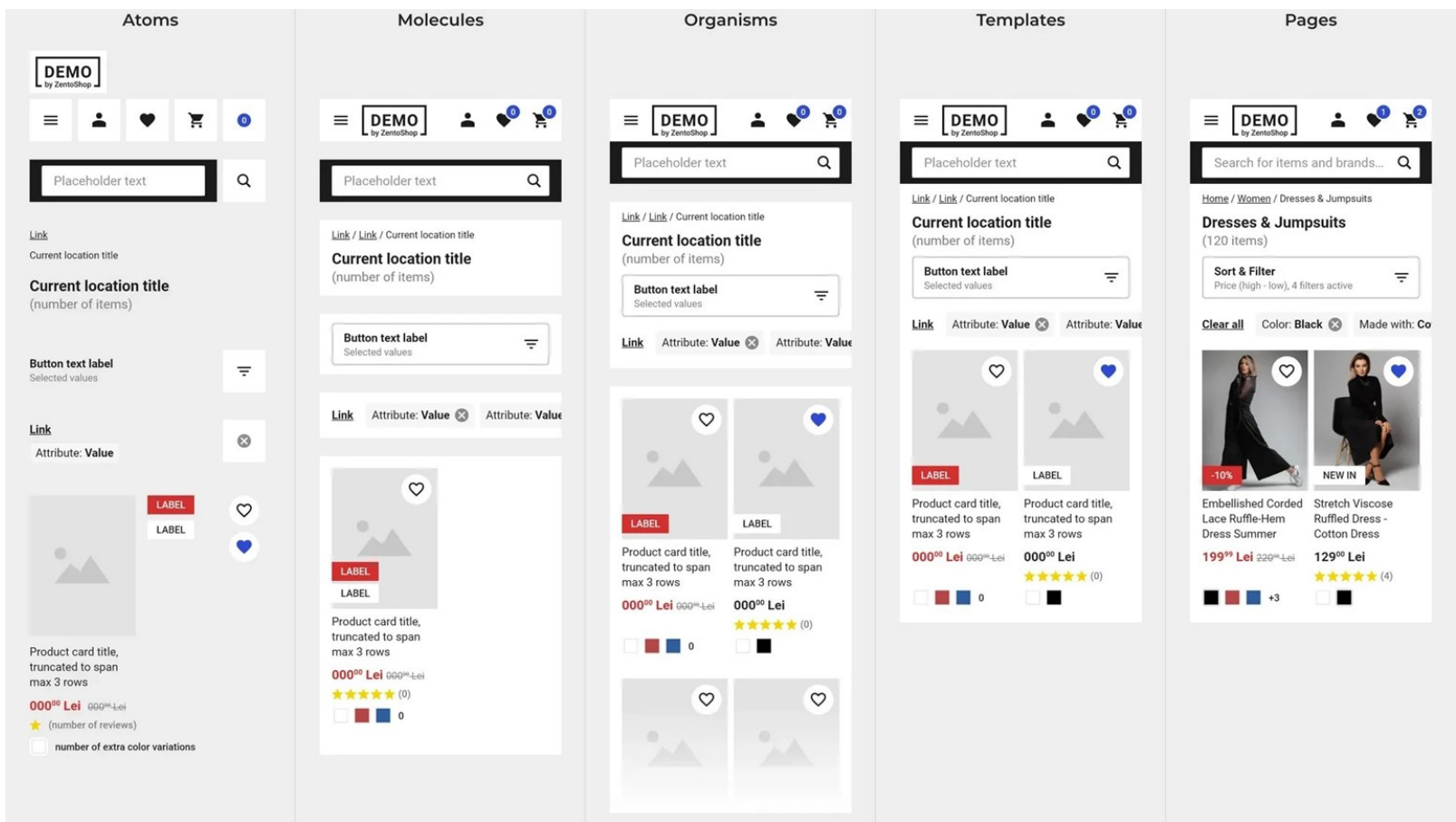
Thoroughly test your design system to identify and fix any issues

- **Be flexible:**

Adapt your design system to meet the changing needs of your organization and users

Atomic Design - Conclusion

Atomic Design is a powerful methodology for creating consistent, reusable, and scalable design systems. Allows you to create user interfaces that are easier to develop, maintain, and update

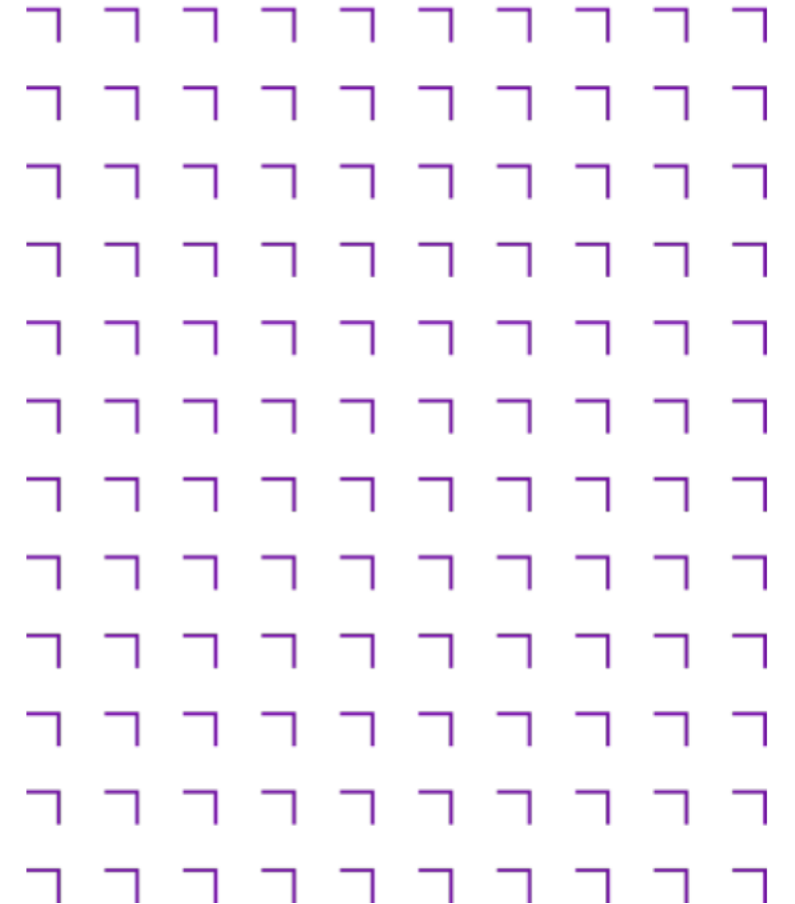




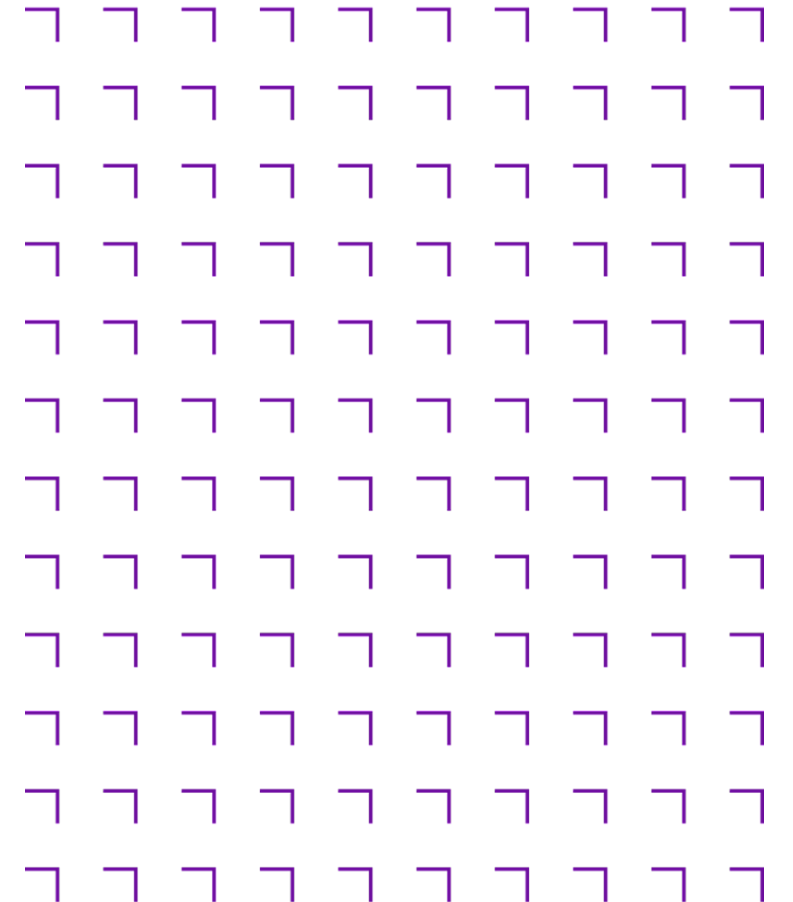
Activity

Breakout

- Join a breakout room
- Continue working on the unit 6 exercises
- You have 35 minutes
- Lecturer will visit each room in turn, etc...
- Will start next topic on the hour



Building website using GitHub Pages and Astro



Resources

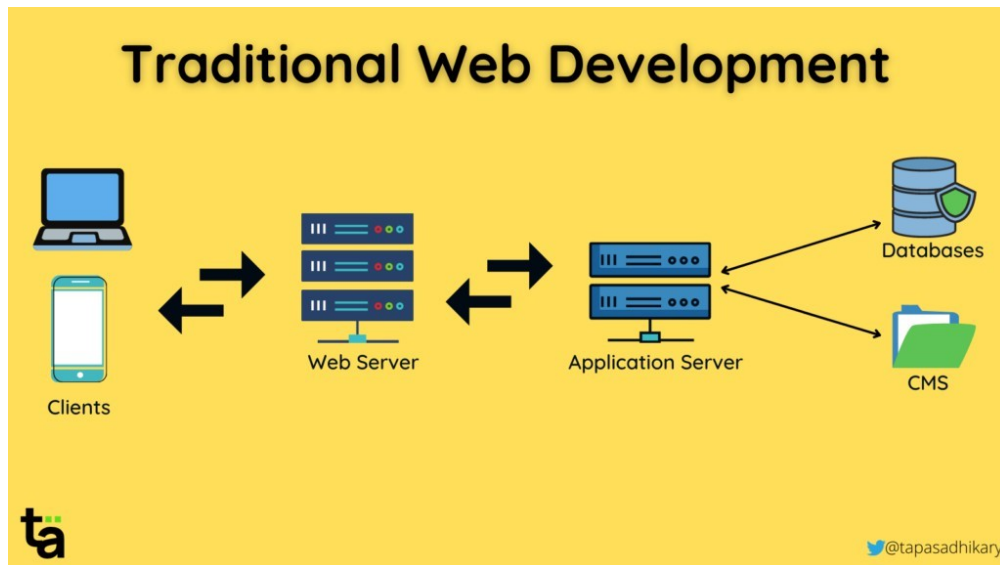
- <https://24ways.org/2014/what-it-takes-to-build-a-website/>
- https://en.wikipedia.org/wiki/Content_management_system
- <https://bugfender.com/blog/top-static-site-generators/>

Content Management System

A content management system (CMS) is computer software used to manage the creation and modification of digital content (content management)

Two major components: a content management application (CMA), as the front-end user interface that allows a user, even with limited expertise, to add, modify, and remove content from a website without the intervention of a webmaster; and a content delivery application (CDA), that compiles the content and updates the website

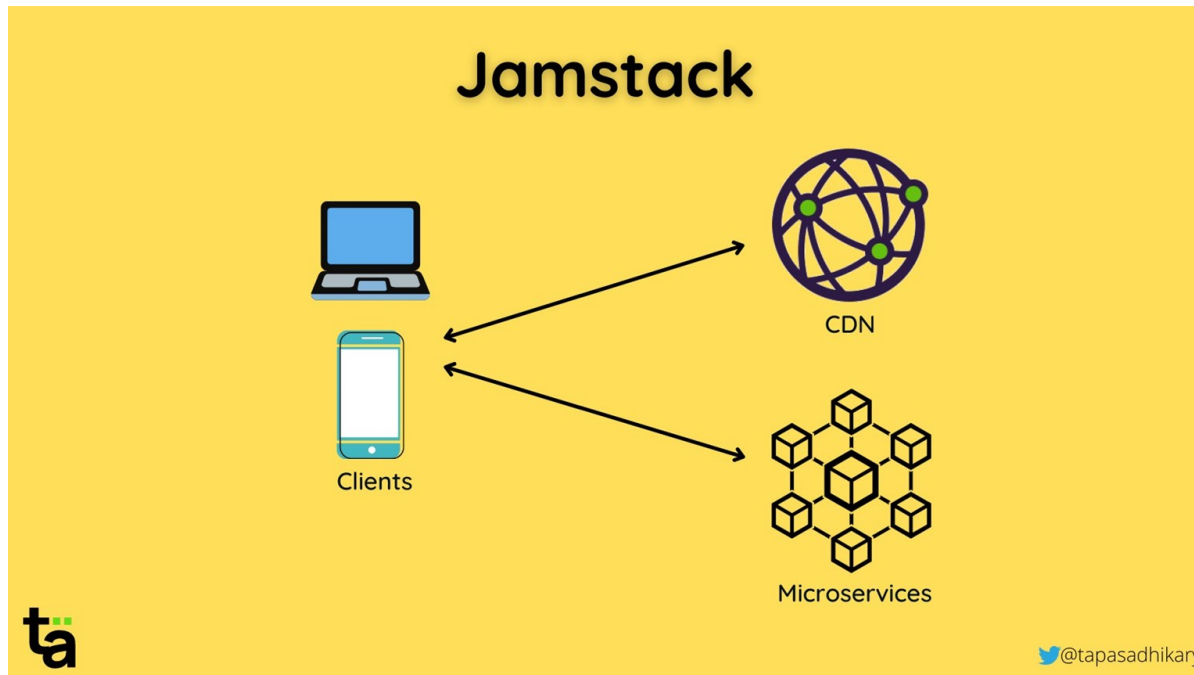
Examples: Squarespace, Contentful, Wordpress.com, Webflow, Ghost and WIX



Jamstack

In late 2016, Jamstack arrived as an architectural concept. The architecture is based on three key features:

- Client-side JavaScript
- Reusable APIs
- Prebuilt Markup



Core Principles of Jamstack

- **Static Site Generation:**

Create pre-built HTML pages during deployment, eliminating server-side rendering and improving page load times

- **Client-side JavaScript:**

Empower the client with JavaScript to handle dynamic interactions and user experiences

- **Reusable APIs:**

Leverage APIs to fetch and display data from various sources, enhancing scalability and flexibility

- **CDN Delivery:**

Utilize Content Delivery Networks (CDNs) for faster content delivery and global reach

Benefits of Jamstack Architecture

- **Enhanced Performance:**

Blazing-fast page load times due to static pre-rendering and CDN optimization

- **Scalability and Agility:**

Easily scale up or down based on traffic demands, and adapt to changing requirements

- **Security and Reliability:**

Benefit from the inherent security and reliability of static sites

- **Cost-Effectiveness:**

Reduce infrastructure costs and simplify maintenance with a serverless approach

- **Developer Productivity:**

Streamline development workflows and focus on building rich client-side experiences

The Role of Static Site Generators (SSGs)

- Static Site Generators are essential tools for creating pre-built markup in Jamstack architecture
- They automate the process of generating static HTML pages from templates and content sources

Content Management Systems (CMSs) for Jamstack

- Jamstack sites often utilize headless Content Management Systems (CMSs) that are specifically designed for working with static sites
- These CMSs focus on managing content and interacting with APIs, leaving the rendering and presentation to the client-side

Jamstack in Action: Real-World Applications

Jamstack has gained traction in various industries and use cases, including:

- **E-commerce:**

Powerful and performant e-commerce websites with a seamless user experience

- **News and Publishing:**

Dynamic news sites and blogs with real-time content updates

- **Blogging and Personal Websites:**

Personalized and engaging blogs and personal websites that prioritize speed and performance

- **SaaS Applications:**

Backend-as-a-Service (BaaS) platforms that deliver APIs to power client-side applications

The Future of Jamstack

Jamstack continues to evolve as a leading approach to web development, with emerging technologies and frameworks shaping its future

- **Progressive Web Apps (PWAs):**

Integration of PWA features for offline access, app-like experiences, and native app capabilities

- **Serverless Functions:**

Leveraging serverless functions to handle dynamic tasks and API integrations without managing backend infrastructure

- **Machine Learning and AI:**

Incorporating AI and machine learning capabilities to enhance content personalization, user experiences, and data analysis

Getting Started with Astro (Walkthrough)

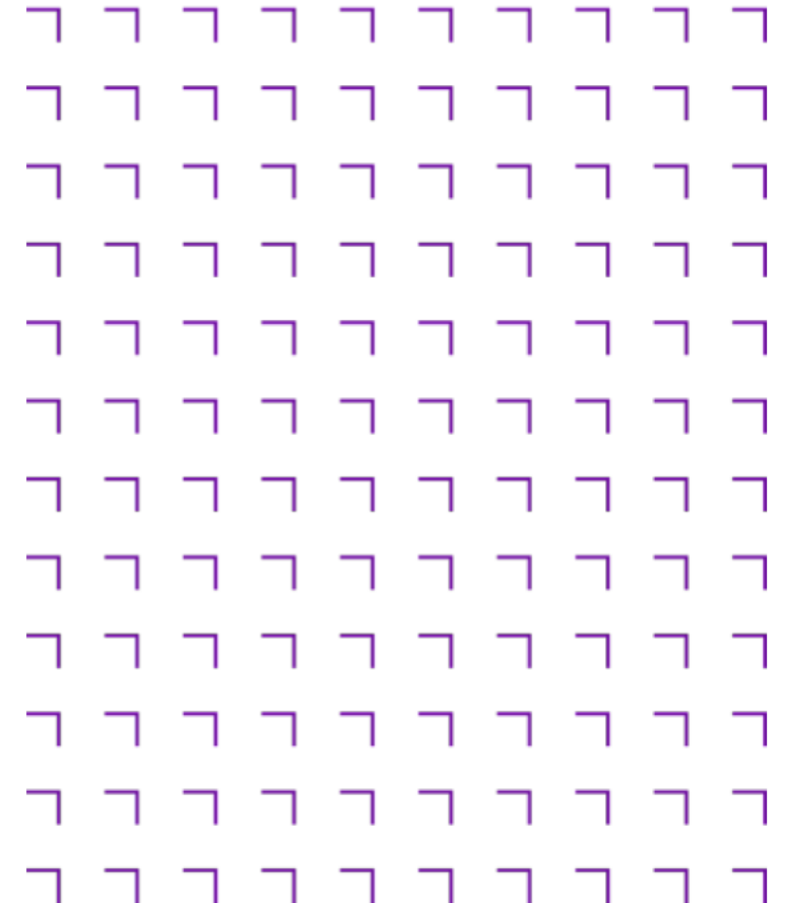
- Installing Astro - <https://docs.astro.build/en/getting-started/>
- Building a simple one page website
- Uploading to GitHub
- Deploying to GitHub Pages

Activity



Breakout

- Join a breakout room
- Continue working on the unit 6 exercises
- You have 35 minutes
- Lecturer will visit each room in turn, etc...
- Will start next topic on the hour



Summary



Completed this Week

- Information Architecture and Site Maps
- Create page components, templates/layouts
- Build site using Astro and deploy to GitHub Pages

For Next Week

- Complete the remaining exercises for unit 6 before next class
- Review the slides and examples for unit 7

