Ex 6:- Demonstration of URL Design

Aim: -

Procedure:-

1. Why do we need URL shortening?
URL shortening is used to create shorter aliases for long URLs. We call these shortened aliases “short links.” Users are redirected to the original URL when they hit these short links. Short links save a lot of space when displayed, printed, messaged, or tweeted. Additionally, users are less likely to mistype shorter URLs.The shortened URL is nearly one-third the size of the actual URL.
2. Requirements and Goals of the System
You should always clarify requirements at the beginning of the interview. Be sure to ask questions to find the exact scope of the system that the interviewer has in mind.
Our URL shortening system should meet the following requirements:

- Functional Requirements
- Non-Functional Requirements
- Extended Requirements

3. Capacity Estimation and Constraints
Our system will be read-heavy. There will be lots of redirection requests compared to new URL shortenings. Let’s assume a 100:1 ratio between read and write.
Traffic estimates:
Storage estimates:
Bandwidth estimates:
Memory estimates:
High-level estimates:
4. System APIs
Once we’ve finalized the requirements, it’s always a good idea to define the system APIs. This should explicitly state what is expected from the system.
We can have SOAP or REST APIs to expose the functionality of our service.
Following could be the definitions of the APIs for creating and deleting URLs:
createURL(api_dev_key,original_url,custom_alias=None,user_name=None,expire_date=None)
api_dev_key (string): The API developer key of a registered account. This will be used to, among other things, throttle users based on their allocated quota.
original_url (string): Original URL to be shortened.

custom_alias (string): Optional custom key for the URL.
user_name (string): Optional user name to be used in the encoding.
expire_date (string): Optional expiration date for the shortened URL.
Returns: (string)
A successful insertion returns the shortened URL; otherwise, it returns an error code.
deleteURL(api_dev_key,url_key)
Where "url_key" is a string representing the shortened URL to be retrieved; a successful deletion returns 'URL Removed'.
How do we detect and prevent abuse? A malicious user can put us out of business by consuming all URL keys in the current design. To prevent abuse, we can limit users via their api_dev_key. Each api_dev_key can be limited to a certain number of URL creations and redirections per some time period (which may be set to a different duration per developer key).
5. Database Design
Defining the DB schema in the early stages of the interview would help to understand the data flow among various components and later would guide towards data partitioning.
Database Schema:
We would need two tables: one for storing information about the URL mappings and one for the user's data who created the short link.
6. Basic System Design and Algorithm
The problem we are solving here is how to generate a short and unique key for a given URL.

a. Encoding actual URL
b. Generating keys offline

7. Data Partitioning and Replication

a. Range Based Partitioning:
b. Hash-Based Partitioning:

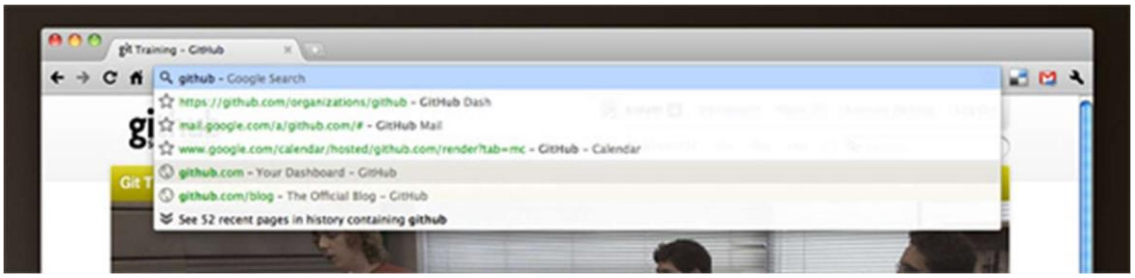
8. Cache
We can cache URLs that are frequently accessed. We can use any off-the-shelf solution like Memcached, which can store full URLs with their respective hashes. Thus, the application servers, before hitting the backend storage, can quickly check if the cache has the desired URL.
9. Load Balancer (LB)
We can add a Load balancing layer at three places in our system:

1. Between Clients and Application servers
2. Between Application Servers and database servers
3. Between Application Servers and Cache servers

10. Purging or DB cleanup
11. Telemetry.

12. Security and Permissions

Output:



1. ASCII-only user generated URL parts (defunkt, resque).
2. “pull” is a short version of “pull request” — single word, easily associated to the origin word.
3. The pull request number is scoped to defunkt/resque (starts at **one** there).
4. Anchor points to a scrolling position, not hidden content.

Result:-

Ex 7:- Demonstration of Web Browser design

Aim:

Description: Design in the browser is basically a concept using HTML and CSS as your primary design tools. Basically the code is writing right from the scratch during each phase of the project. From the clients brief on to the first design draft, to an rudimentary prototype to a finished product. Everything (or nearly everything) takes place in the browser. Rather than spending hours designing pixel-perfect design drafts in Photoshop, designing in the browser allows you to jump directly into the text editor and start shaping your code.

Demonstration:

Tools for Designing in the Browser

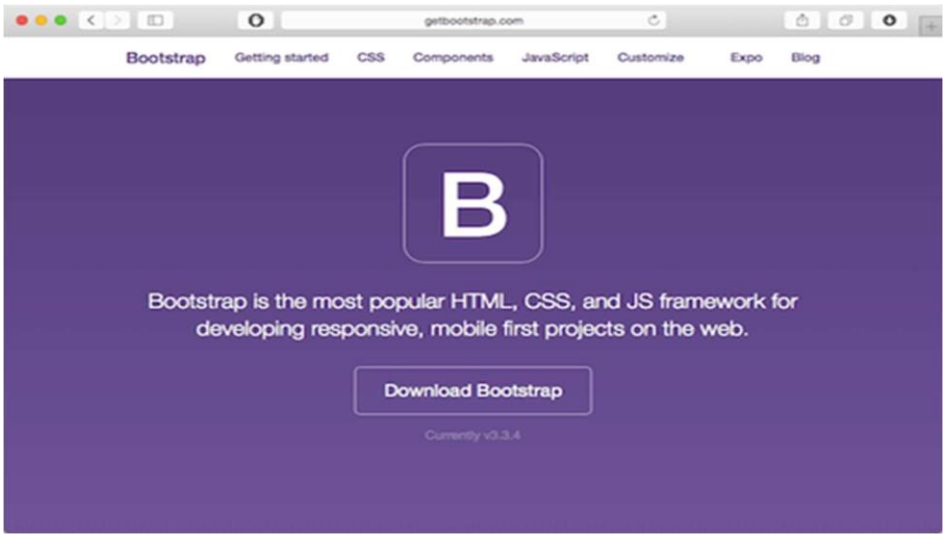
Convinced of the "design in the browser" concept? Want to get right into it? Perfect, here are a couple of great tools which are making your life easier when designing in the browser.

- Get a programming friendly editor

  This might sound a bit strange. But designing in the browser basically means working a lot on the code of your site. The editor will be your friend and go-to tool in many cases. Choose your editor wisely.

- Bootstrap – your front-end framework

  Bootstrap is probably the best known front-end prototyping framework available. Originally designed by Twitter, it's now available to everyone for free. Packed with some great functionality, it supports typography, forms, buttons and some great JavaScript options.



· Style guide

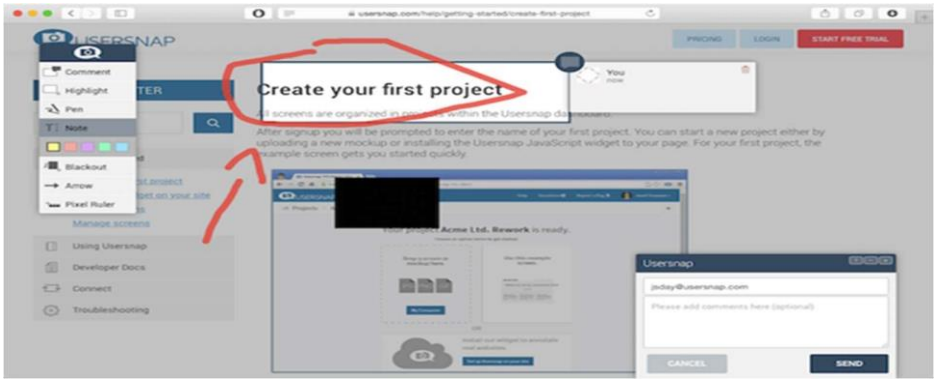
Next, it's about the style guide. It's super important to keep your design and style elements organized. With a style guide in place, design changes are super easy as they will come. And trust me: they will come.

· Chrome Developer Tools

After you created your first prototype in your browser, it's time to review, test and tweak. The best tools therefore are available for free in every Chrome browser (or Firefox if you prefer). With a right click on your site you can start the Chrome developer tools which offer you a broad range of features. You can play around with your code, move styles, edit content and much more.

· Keep track & manage tasks with Usersnap

Since designing in the browser is quite an agile approach, you might wonder how you keep track of change requests, bugs, ideas, etc. I therefore recommend Usersnap, which does a great job making collaboration on website or web app projects much easier. By adding the Usersnap widget to your prototype, every single piece of comment and feedback will be stored in your project overview where you can discuss feedback and track bugs.



Conclusion:

Designing in the browser still requires great design skills. No tool in the world will make up for your lack in design know how. However, the browser becomes more and more your design and development environment. And because most of the design process takes place in the browser, doesn't mean you should give up Photoshop.

Result:

Ex 8 :- Demonstration of Navigate design

Aim:

Description:

Navigation design is the discipline of creating, analyzing and implementing ways for users to navigate through a website or app.

Navigation plays an integral role in how users interact with and use your products. It is how your user can get from point A to point B and even point C in the least frustrating way possible.

To make these delightful interactions, designers employ a combination of UI patterns including links, labels and other UI elements. These patterns provide relevant information and make interacting with products easier.

Good navigation design can:

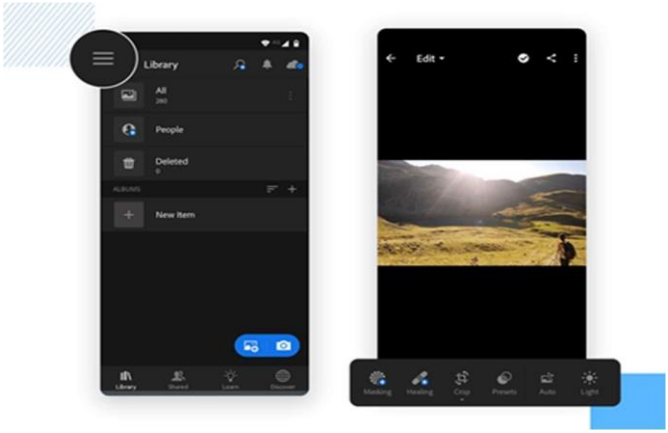
- Enhance a user's understanding
- Give them confidence using your product
- Provide credibility to a product

Implementing common navigation design patterns in UX design:

Many products will use a combination of these mechanisms in their designs because some patterns work better depending on the circumstances at hand.

Hamburger menu

The hamburger menu is often found on mobile, although it is increasingly becoming popular with desktop. The hamburger menu icon is 3 lines and can be clicked or tapped to reveal more navigation options.



Hamburger menu (left) and Tabs (right)

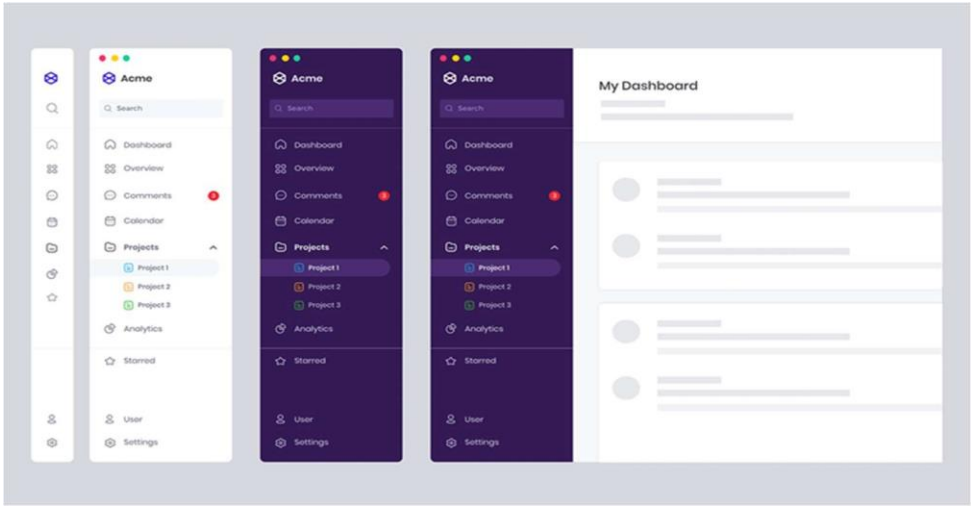
Tabs

Tabs are a popular navigation pattern and are commonly found on mobile devices. They're can be found on the bottom or top of the screen. Because you can only fit so many tabs at the bottom of your screen, you'll usually find the most important screens in a tabbed navigation.

Vertical navigation

Usually found on the left-hand side of screens, vertical navigation takes up generous screen space but displays a list of global navigation links and can include primary, secondary and tertiary navigation levels.

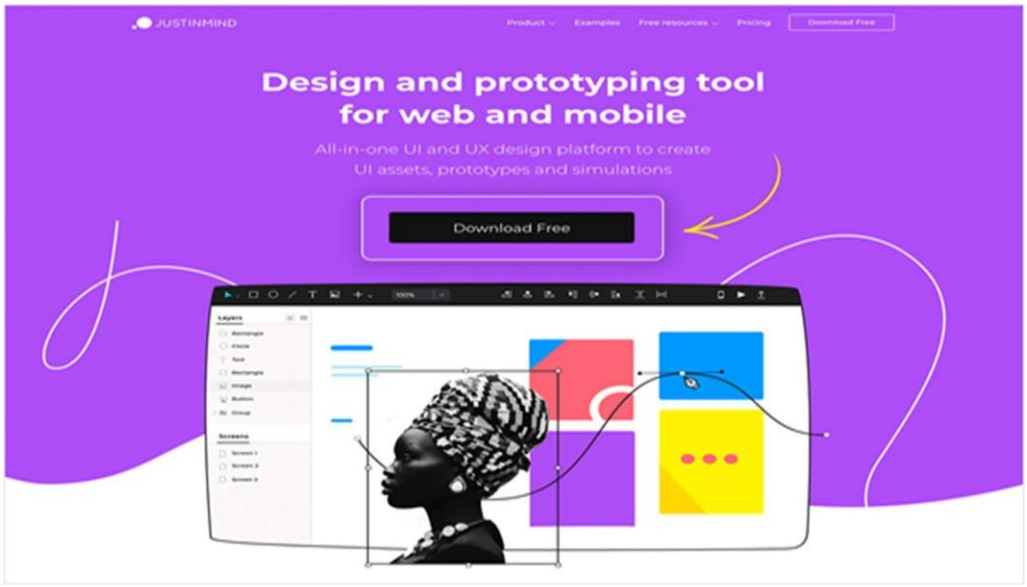
These are handy when the user already has a lot to digest in the screen. That means that in products that involve a lot of data, such as dashboard design, hiding the navigation options can be the best way to lower cognitive effort.



Vertical navigation menu

Call-to-action button

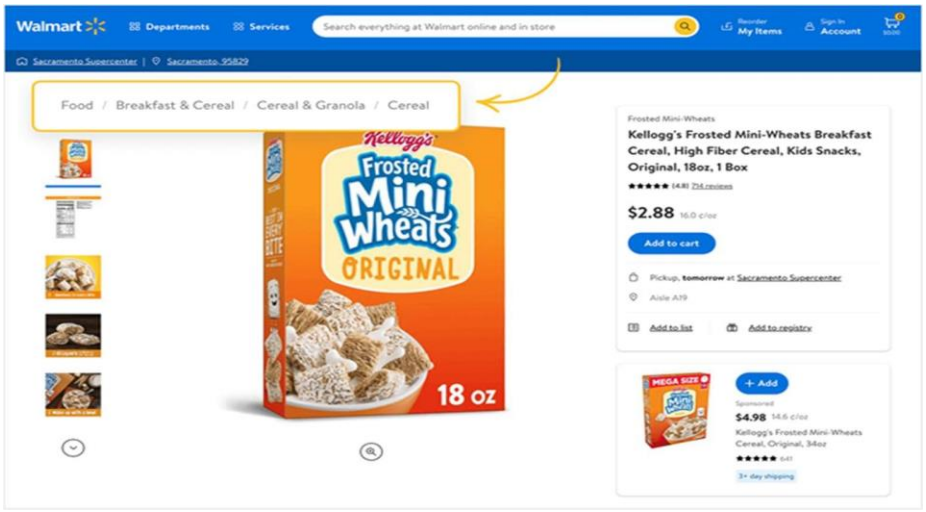
Call to action buttons are used to persuade, motivate and move your audience into an action whether it’s a sign up, a purchase or a download. They are usually given prime of place on websites and must be noticeable. Discover how picking the right UI colors can boost your conversions through the roof.



A prominent call to action button

Breadcrumbs

Inspired by the story of Hansel and Gretel, breadcrumb navigation (or breadcrumb trail), is a secondary navigation system that shows the user where they are in the system.



Conclusion:

There is a lot that goes into perfecting navigation design and getting it right means it usually has to go unnoticed. By aligning your user goals, content strategy and navigation design, you'll be able to create a cohesive and consistent user experience that your users will love.

Result: