

Colour model

Colours have moods & colours affects brain & change / apply mood of colour to person like

red \Rightarrow love, energy, intensity

yellow \Rightarrow Joy, intellect, Attention

green \Rightarrow Freshness, safety, growth

blue \Rightarrow stability, trust, serenity

Purple \Rightarrow royalty, wealth, femininity

Yellow is attention as it is ignore the yellow is hard

\Rightarrow Colour wheel

two corresponding colours \Rightarrow analogous

• good for navigation bar, body

\Rightarrow Opposite pair of colour \Rightarrow complementary colour pallet

look quite cache -

\Rightarrow It make things pop so it seek more attention, but complementary is not good for body it is very eye cache for logo & like stuff.

\Rightarrow Col

of adobe color

Colourhunt ready made colour pallets

① Serif font

Serif font has feet like & V, F
→ for feet they called it serif

② Sans serif

It has more right angle &
→ It don't have feet
→ sans serif is good for feet
→ sans serif is very eye-catchy.

⇒ multiple fonts in a website is look weird & bad at same time

⇒ max 2 fonts are to be used & it should create contrast.

Manage the attention

① Hierarchy

The most important information will get higher font-size & other corresponding to the weightage in document

if

→ All colour same is does not look attractive

∴

→ multiple colors look good but not more than 2-3 colors

⇒ Bigger item drew the first attention

⇒ Layout

Blocked text containing pictures, texts attract majority of people

→ for content there is a good line length
See we Wikipedia many of us avoid using Wikipedia as it is so much lengthy

40-60 character per line is attractive

⇒ Alignment

① Same alignment of text & heading look good & it is eye catchy also

⇒ Visualising grid lines are good idea in order to see the alignment of object in your design

⇒ White space

Spacing things, means more space things with more amount of space suddenly grab attention than things which look quite compact

→ White space can make design minimalist & suddenly grab more attention

⇒ Design is more important for attention & we should know target audience

UX user experience design

① Simplicity

keeps things as simple as possible

② Consistency

keep all things rolled up & keep consistent

③ Reading pattern

from most of time when viewing people strive to read in F Pattern so always try to keep design in F way to it catch more content

→ Z pattern is also good & we can use it very much

④ All platform design

website should support all devices like tablet pc & you have work according to it.

⇒ so rendering of screens of multiple devices

→ Scaling is also matter

& Always try to do test & modify meth & use suggestion from people

avoid dog pattern like click bait

→ you can use dog pattern to make more of your website
so can able

Page No. _____
Date _____

→ checkbox is also example of dog pattern

printing of dog pattern is short & good
Express JS is combination of Node.js & MongoDB

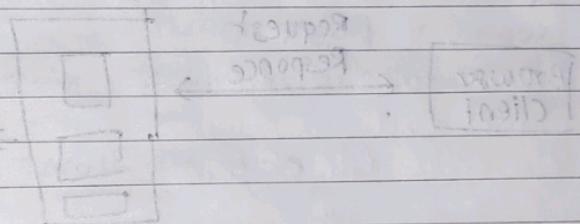
ZIP now not encourage not find in search and
Anonymity

not in offices. many companies not set to
checkbox

→ we can apply to create a secure environment

see to

ZIP



let

zip will do

not do

(zip) (zip = zip = zip)

and (zip) (zip = zip = zip)

and (zip) (zip = zip = zip)

(zip) (zip = zip = zip)

(zip) (zip = zip = zip)

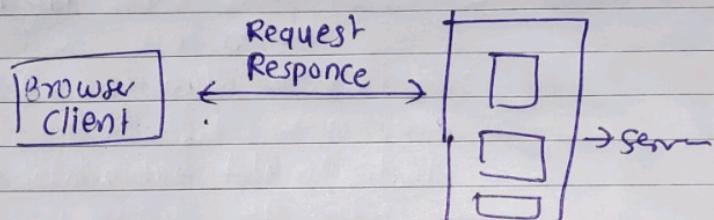
#Express JS

node \Rightarrow node is great source to do anything which is related to hardware with javascript only

but express is built for specifically for web app development

\Rightarrow But for backend express is very specific for the express js

\Rightarrow we can able to create a server directly with express js



Browser & server talk in language of response request

so with listen
see code

$\$ \ const \ Server = require('express');$

$Server = Server.listen() \rightarrow$ this will start
server

$\$ \ Server(115) \text{ in (Port 192)}$



So upto it we create a socket at port 3000

```
const one = require('express')
one app server One. express() create
exp() a ser
port = 3000;
```

server.listen(port)

So If you run this snippet only you will get
only the server listening but not get any
response for that we have to use

```
serve.get("location", function to be called)
          ↑
          after get request
the get request is request which
serve receive & it give response
```

localhost:3000 → local
It is not here

) function(request, response) {

console.log(request); -

// It will print request you get from browser

& response.send("msg you want to send");

}

and that's it

⇒ You can able to give multiple response
to multiple location

get & listen are request which help
to do conversation b/w host & server

⇒ You can send wrapped html under get & send
as our respon

app.send(" <h1> Hello world </h1> ")
response Inside double quotes

⇒ use nodeman so that you don't have close &
restart server again as it take lot of time
to run Close server.js file

app.send(" / ")

route different routes different values
for get

⇒ Response with & html file

to send entire file of html

response.sendFile("location of file")

-- dirname in send file will give location of
(current file folder)

-- dirname it is like pwd in terminal

So we can map our file location
properly

--dirname is a Javascript keyword for
pwd

so send like

response.sendFile(`--dirname + '/index.html')`

① Cannot post index.html

action tab in a form will send received data to that specific location through data object & variable name will be name you specified in input

⇒ So study how to deal with post requests

```
app.post('/1', function(req, res){})
```

res.send('thank you')

status code 200 = OK

so we have to deal with outputs

so we will use body parse npm package

& now we have add parser packet in js file:

as

```
const f = require("body-parser")
```

then we can parse request & get info out of it

-app.use('body')

Code

```
const bp = require('body-parser')
app.use(bp.urlencoded({ extended : true }))

app.use(bp)
```

& now we can use body parser

& now this will activate your route &

So if we use

req.body you will get a JS object
having value written from the html & posted
to you via post

& After that we are able to parse 'get access'
to data

so we can tap value from it via .dot
operator

req.body
is parsed data part