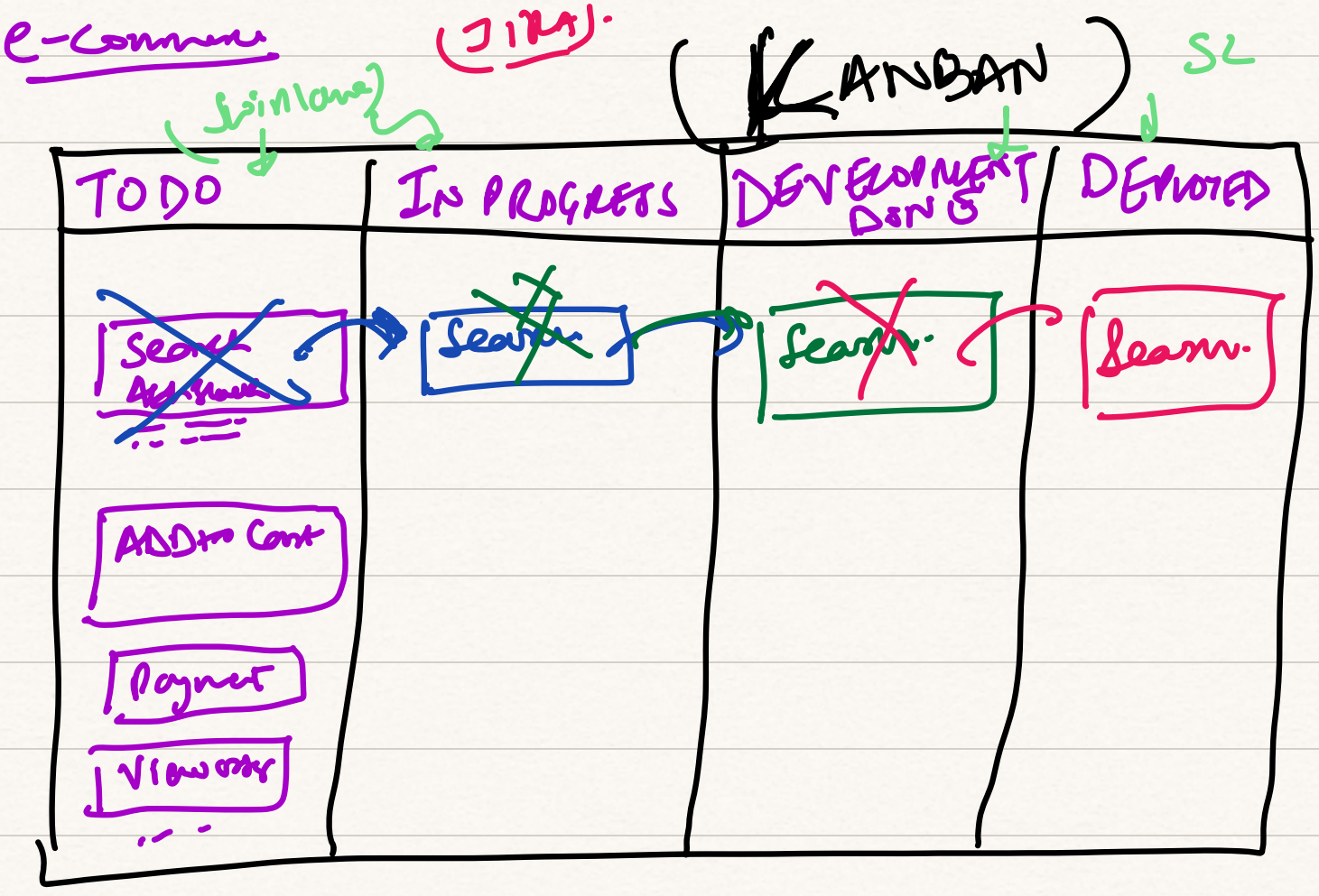
28. KANBAN BOARD

Agenda :

- Description of the project (across 4 classes)
- Requirement & Features of the project



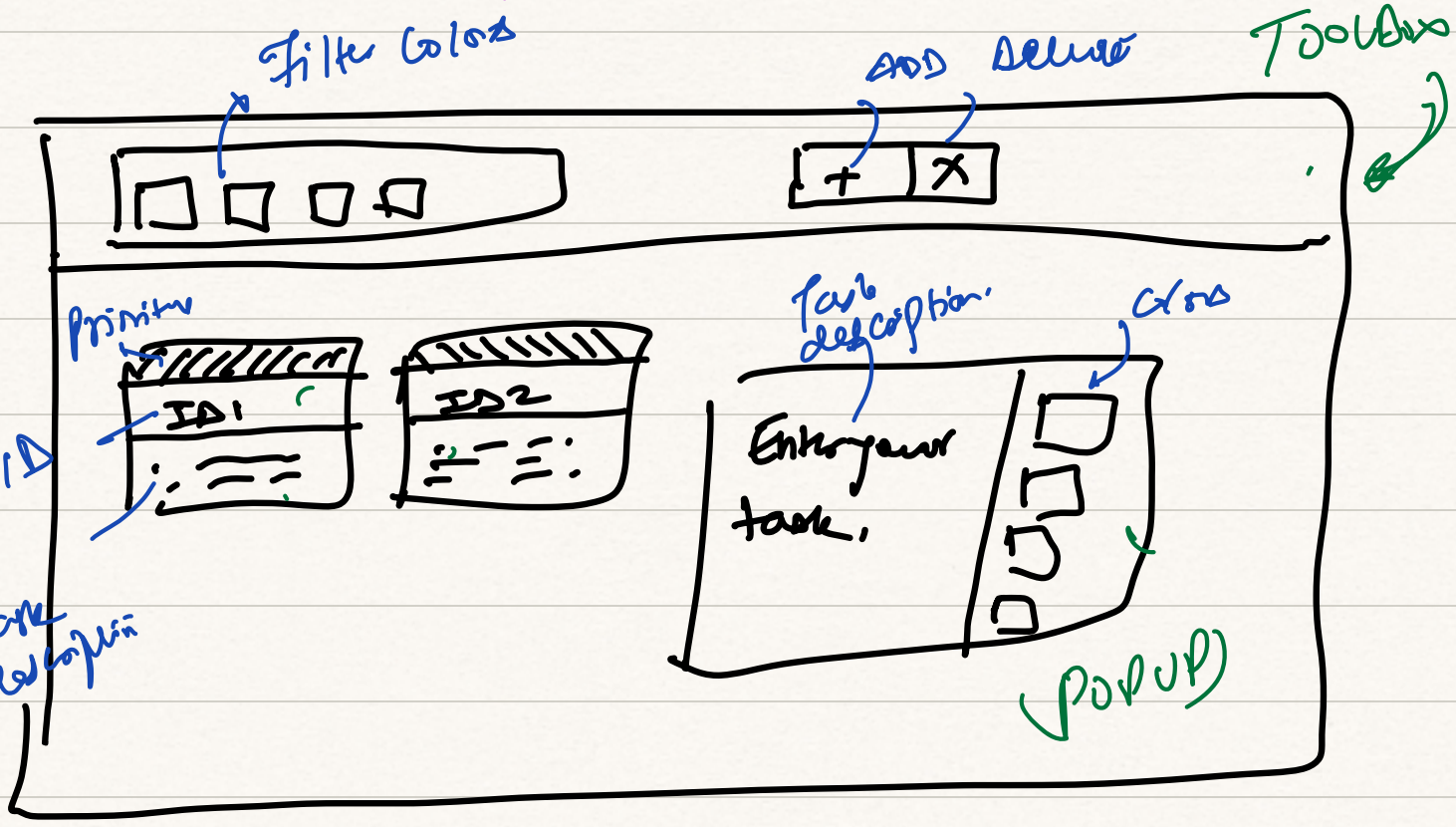
Kanban → • Task management Application
• Enhances productivity

- Give tracking info to stakeholders.
- Visual format for organizing tasks.
  - Stage / priority / Assignee / description / estimate of task

FEATURES of the project :

0) Board / UI Elements creation.
1) Dynamic task addition
2) Filter the task based on Color coding
3) Edit task descriptions (lock/unlock).
4) On refresh data persist [LOCAL STORAGE]
5) Task deletion
6) Change priority color of a task
7) Unique task identification [id for each task]
8) Responsive design / UX should be good.

Wireframe of the project :



Color Priorities :

Pink → TODo (task to be done)
Green → In progress
Blue → To be validated (Dev Done)
Black → DONE / DEPLOYED.

Class 2 :

Agenda :

- Modal Popup with toggle
- Ticket generation
- Adding task, generate ID, Colour to generated ticket
- Ticket Removal.

Modal : Show / Hide

<boolean>

showPopup → true

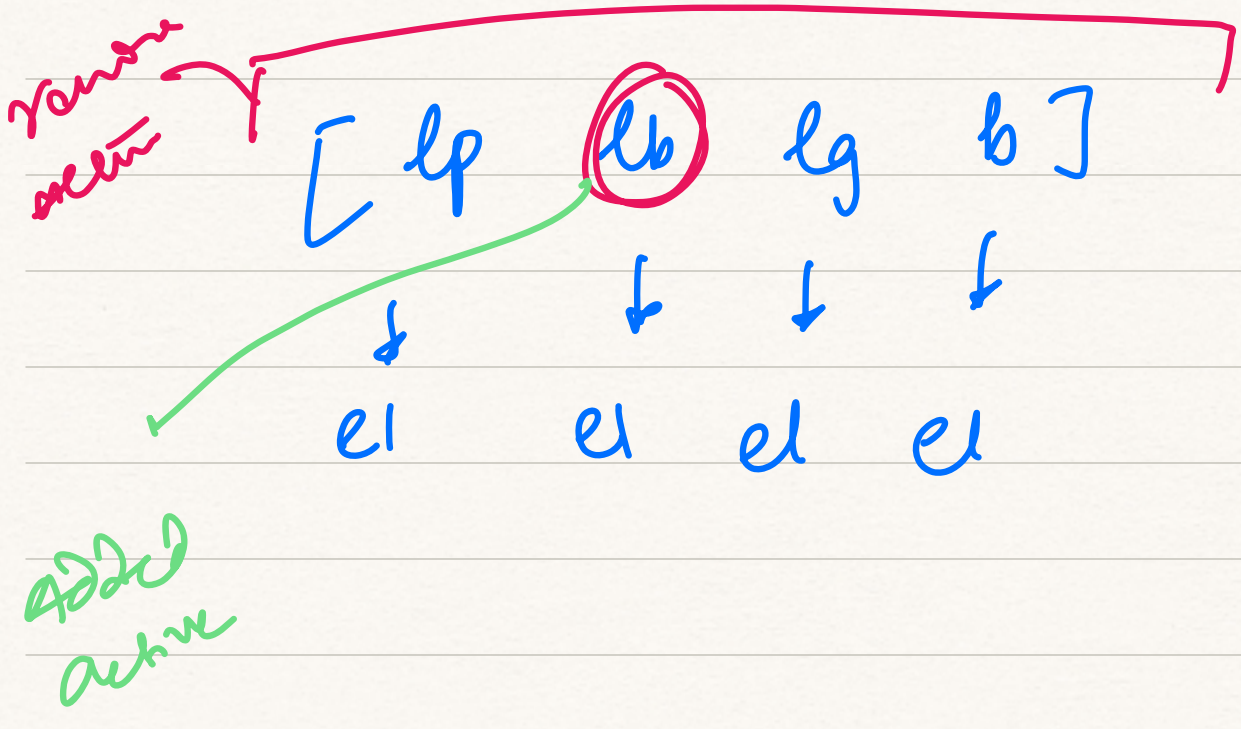
→ false

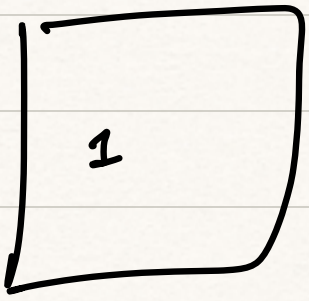
x

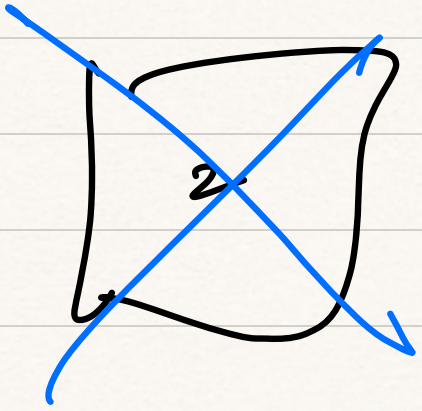
(by default).

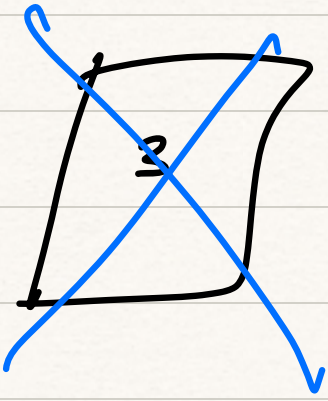
Requirement : (To add task color, description, ID)

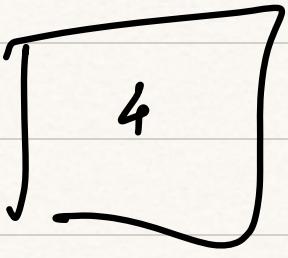
- Choose a color & that color should be applied to the generated ticket..
- Generate & add IDs
- Task description ✓

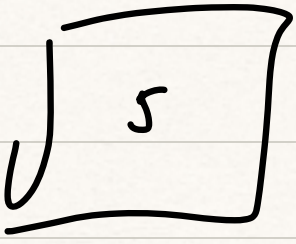


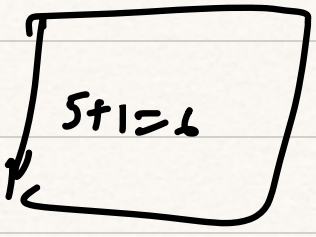

1

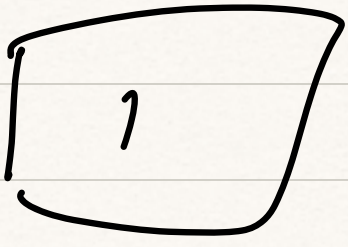

2


3

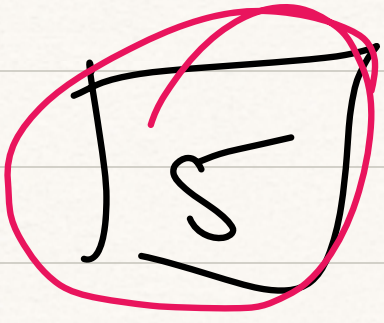

4

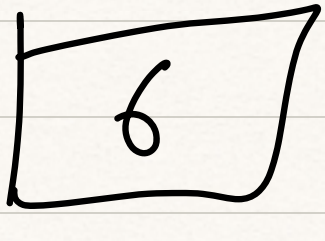

5

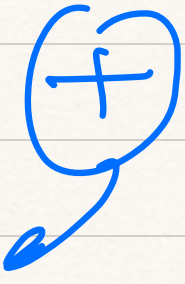

5+1=6

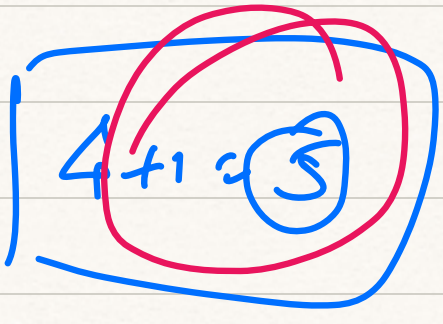

1


4


5


6


+


4+1=5

Next Class (17/9/24)

- Handle lock
- Filter task on color
- Implement Local Storage.

* KANBAN-3 :

✓ Locking mechanism
✓ Color change of priority in a task
• Filtering of color in a task.
. .

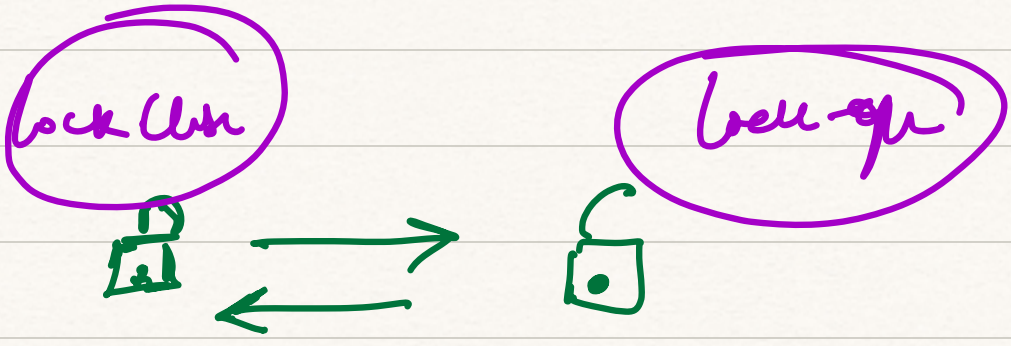
• State change •

Locking mechanism

show/hide

① Lock ⟷ unlock icon

② Clicking unlock, should edit the task box.



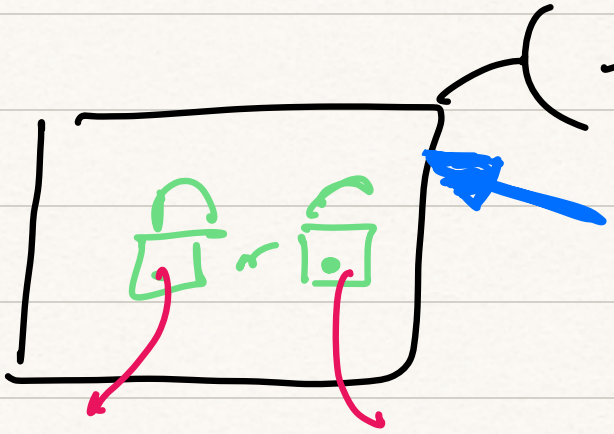
Ways to impl.

1. Flag variable

↳ if (lockOpen) — true → fa-unlock

false → fa-lock

2. Add dynamic CSS.



(ticket-lock div)

If this div container contains 'fa-lock' class,

→ Remove lock class

→ Add fa-lock-open class.

fa-lock

fa-lock-open

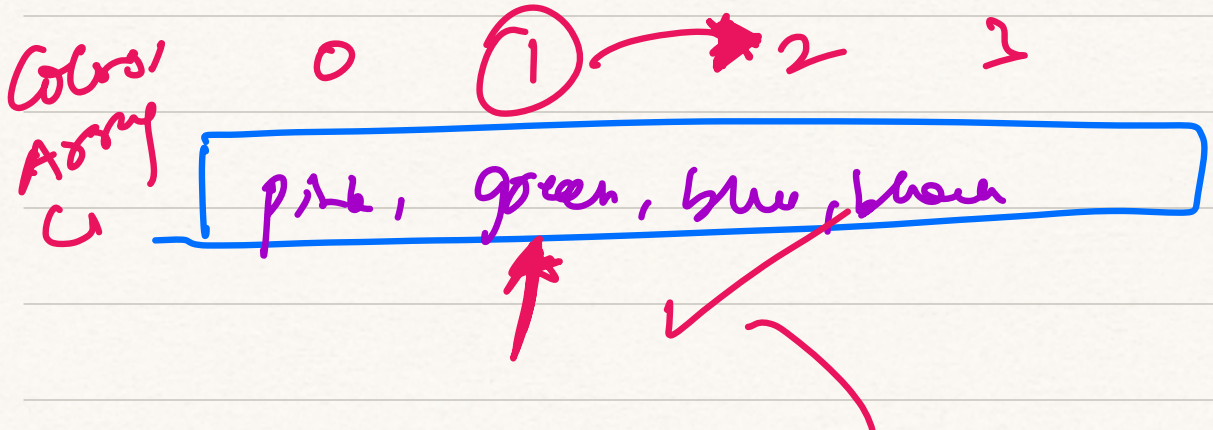
OK

if div container has unlockClass,
- Remove unlock/open class
- Add lock class.

* contenteditable attribute makes anything in HTML editable.

or

* Not all properties in HTML can be edited.

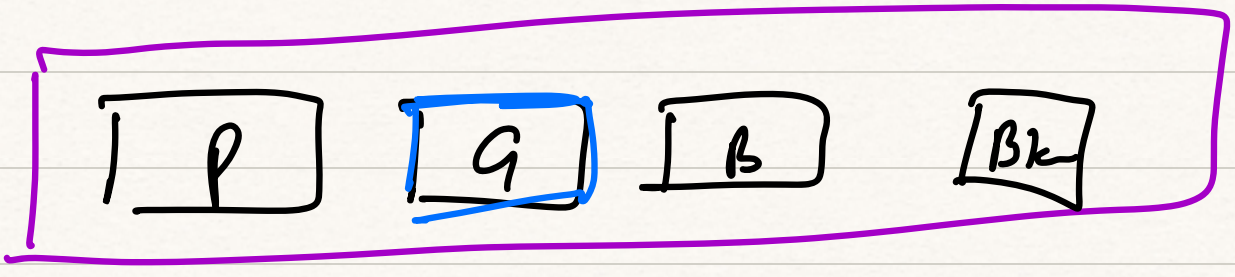


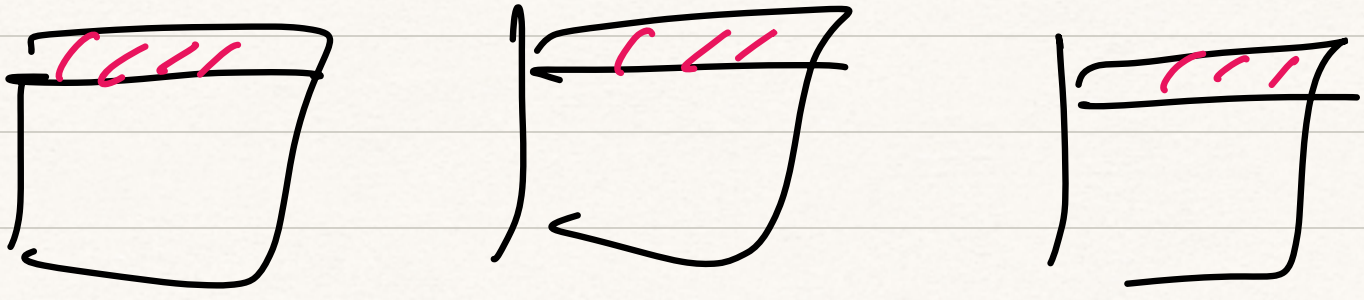
(1) Find current color which is clicked

(2) See its index in colors array

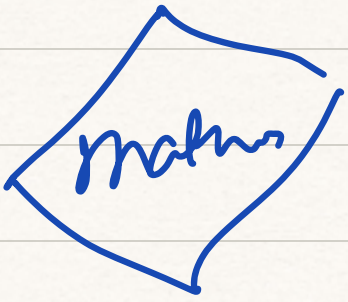
(3) Find next color (index+1)

(4) Update ticket with that color.





Colors array meaning ticket



Show ticket (display block);

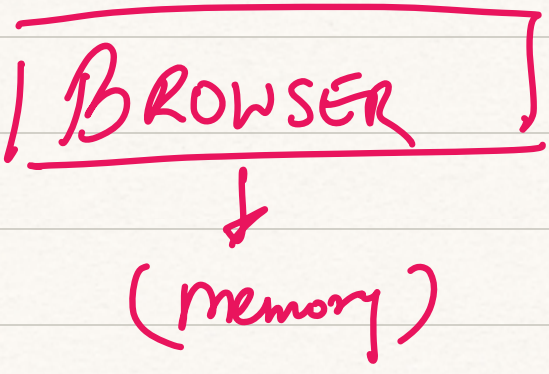
Hide ticket (display none)

KANBAN-4 :

Agenda :

+ Local storage
+ Deploying our code to github pages
  - Leftover find touches as well.

Local storage :



- Web storage mechanism that allows websites to store & access data right from the browser
- No expiry date. (manually clear data)

Window

- Data persists even after browser is closed.

BROWSER/WEB STORAGE → Storing data locally within a browser.

Storage Mechanism in browser

1. Local Storage
2. Session Storage
3. Indexed DB

Cookies (4KB) — set by server
↓
(transfer data)

LS	SS
5MB	5MB
• No expiry data	• When browser window is closed, SS gets deleted
• User data, that we know won't change very often.	• Banking transactions.
AMAZON.IN AMAZON.IN TAB1 TAB2 (LS1)	• Different for different tabs in browser TAB1 TAB2 ↓ ↓ SS1 SS2

Usage of LS

- ✓ Create — C
- ✓ Read — R
- ✓ Update — U
- ✓ Delete / Clear — D

- **Set Item (Add / Create).**

```
localStorage.setItem('key', 'value');
```

- **Get Item (Read)**

```
localStorage.getItem('key'); // 'value'
```

- **Delete Item**

```
localStorage.removeItem('key')
```

- Delete / Clear all LS.

localStorage.clear();

Constraints / Limitation of LS

① Storage Limit (5 - 10 MB)

② Synchronous → LS operation will block the main thread while any operation is performed.
- Performance may be affected.

③ Security - • No server involved
- Everything happening locally in browser [JS]

- XSS [Cross site Scripting]
- We don't store sensitive

info in LS. (eg. User session token).

4) No expiry → Manually clear it.
5) Both key/value have to be strings. (can't save object/Array)

LS in KANBAN →

1) Create Ticket → Save tickets in LS

localStorage.setItem(k,v)

2) Update ticket →
   - Color
   - task

[getItem, setItem]

3) Delete ticket → Remove ticket from LS

[removeItem]

ticket array

[{}, {}, ...]

{
id:
color
task
}

createTicket

newTicket
i will
push to
[ ]

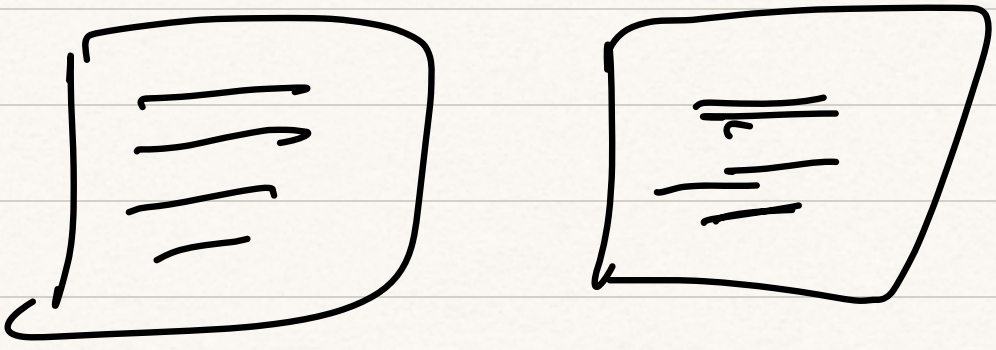
createTicket() { id color task }

ticket array
[= =]
q
UI

LS → [{}, {},]

REFRESH / Close Browser

+ &



init ( ) → Read from LS

NO

YES

for each entry
Show it to

the UI.

Retrieve LS tickets []

{} → CreateTicket()

↓

Show (Print) it to HTML