Date : ______

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

Java

1) oops Concepts < Inheritance / → Polymorphism < overloading / overriding / Abstraction < interface / abstract class

2) exception Handling

3) String

4) Collection

5) Keywords, static, final, Access modifiers.

6) Java 8 streams & lambda expr^n.

xpath Axis / dynamic locators.

Selenium → Locators, → Alerts, windows, iframe, webtables

{ WDr methods
waits
Excep^n In Selenium & Handling (stale element exc^n, not clickable ex^n)

→ Any challenging scenario automated?

Advance: actions, Grid, Java Script Executor

click / scroll

→ TestNG

RealTime Scenarios

Framework:

⇨ Know JD from Recruitement spoc:
details → skill sets: Java, Selenium, BDD, SQL, Protractor

Date:

MON □ TUE □ WED □ THU □ FRI □ SAT □ SUN □

① Greetings

② Introduce yourself (Name, Designation, my colleague)

③ Take Intro (educational background, Tech skills)

④ Read the CV (Ask questions around the Automation things mentioned in cv)

⑤ Check ~~@~~ Learning skills, Thought process when he/she Solves the problem. give hint

⑥ Don't ask direct question available on internet (Palindrome, reverse string, Prim No, fibo.)

⑦ Create your own questionaire, play around the concepts

⑧ if candidate is not good in coding Don't waste time. end it gracefully.

⑨ We can train a person for a Tool but we don't want train them on coding skill. if a person has coding skill he can learn a tool easily.

⑩ What ever questioned on that basis put Strong areas, weak areas. in more detailed ~~form~~ in points
→ if anything else like personal aspiⁿ, commⁿ skill etc weak, strg can also be put in overall feedback, Attitude etc

Overall feedback
1) StrongA →
2) WeakA →
3) Design Evaluation
4) Overall Rating
5) Note/other points.

⑪ you can give candidate your laptop & ask to find the solution of a problem (just check how he approaches to solⁿ what he types, coz we also do the same most of the time copy/paste.)

check smartness

⑫ level of questions should vary as per the designⁿ/exp.

(3.5) yrs if evaluating for SSE 1 level above que. from SE.

1,2,3 → SE
3-5 → SSE (3 to 5)
5-7 → TL-I (5 to 7)
7-9 → TL II (7 to 9)

Avicore → Client Interview Strategy

Date: evcore

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

SSE ⇒ should have Idea about framework creation, if not created any, s^m good idea

[6+1] ⇒ TL I
manual^n
→ Auto

TLI ⇒ should hv frame creation exp or involvement in the same
→ why the f/w is suitable?
Individual Contributor
⇒ metrics used / share with clients.
→ Best Automation practices.
→ Team-Handling exp, Team skills
→ ~~How you Keep urself up to date, blogs, videos~~
→ Latest, just trend in testing (Katalon)
⇒ should be able to work independently & handle client calls.

Approach
To autom^n a manual proj from scratch

Good Look
personality
Thought Process
Smart.
+
Tech ok

Awareness latest trends

for Auto → SE
QA → TLI
metrics
How you keep up to date
Teamskill
Best practices

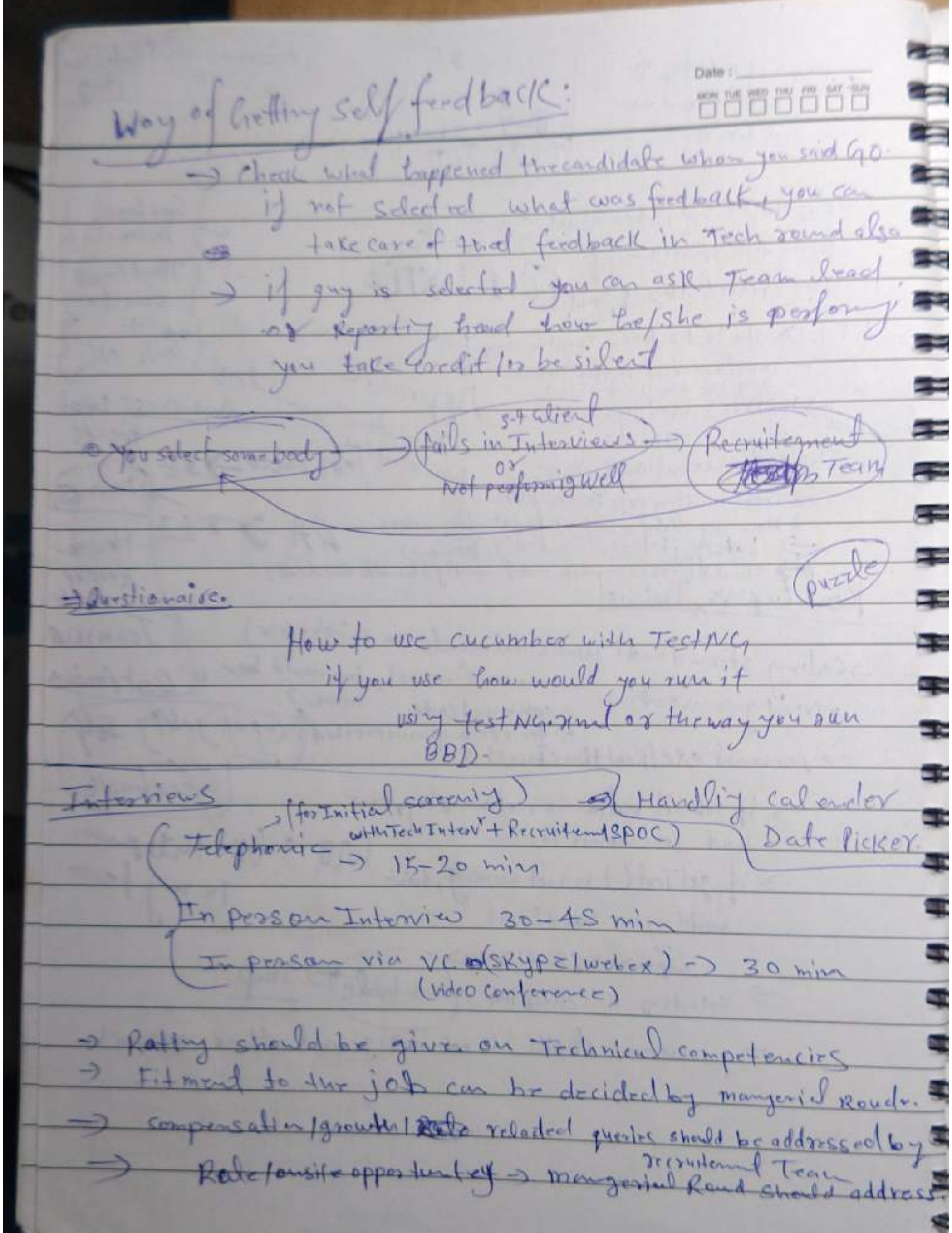
Real Exp vs Trained

→ Coding standards (→ break statements in switch-case)
→ when cond^n is met, loop should have break.)
→ Naming convention
→ optimised code.
→ for a prob. generic method

Way of getting self feedback

→ personal exe / Real time Issues →
→ ~~if submit button is not working, how~~ wud you submit it.
→ if getText() is not working, how would you handle it
<span> xyz </span>
→ selecting a checkbox in web table → Imp.
→ element.click() is not working what other approach

Will I ~~take this~~ person in my team?



Date : ____________

MON □ TUE □ WED □ THU □ FRI □ SAT □ SUN □

Way of Getting self feedback:

→ Check what happened the candidate whom you said GO.
if not selected what was feedback, you can take care of that feedback in Tech round also

→ if guy is selected you can ask Team lead or Reporting head how he/she is perfomy you take credit /or be silent

You select somebody → fails in Interviews (3-4 client) or Not performig well → Recruitement ~~Head~~ Team → You select somebody

puzzle

→ Questionaire.

How to use cucumber with TestNG,
if you use how would you run it
using testNG.xml or the way you run BBD.

Interviews

Handlig calender
Date Picker.

Telephonic → (for Initial screening) with Tech Interv^r + Recruitement SPOC) → 15-20 min

In person Interview 30-45 min

In person via VC (Skype/webex) → 30 min
(Video Conference)

→ Rating should be given on Technical competencies
→ Fitment to the job can be decided by mangerial Round.
⇒ compensation/growth/~~Rate~~ related queries should be addressed by recruitement Team
⇒ Rate/onsite opportunity → mangerial Round should address.

Java Concepts

Date : ________

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

→ What are opps concepts (Encapsulⁿ, Inheritance, polymorphism, Abstraction)

→ Access modifiers rule in method overriding, what are AM?

→ Can we override or overload static(or main) methods?
(No, but) (yes) & final ✓

→ Can We override or overload constructors, what is constructor?

→ Diff b/w final & static

→ Can we override or overload final method.

→ Exception rule in method overriding

→ Can we have private constructor → yes

→ final, finally, finalize()

→ List, Set, Map

Java 8 forEach() Examples: < Present in Iterable class 1. of
present in Stream class.

① forEach of Map →from Iterable class

Date: ______

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

```
hm.forEach((k,v) → sop("Key is:"+K+"value is:"+v));
or hm.forEach((k,v) → {
                 sop(——);
                 if(k.equal("Java"))
                     sop("Hello Java");
              });
```

② forEach of list →for Iterable

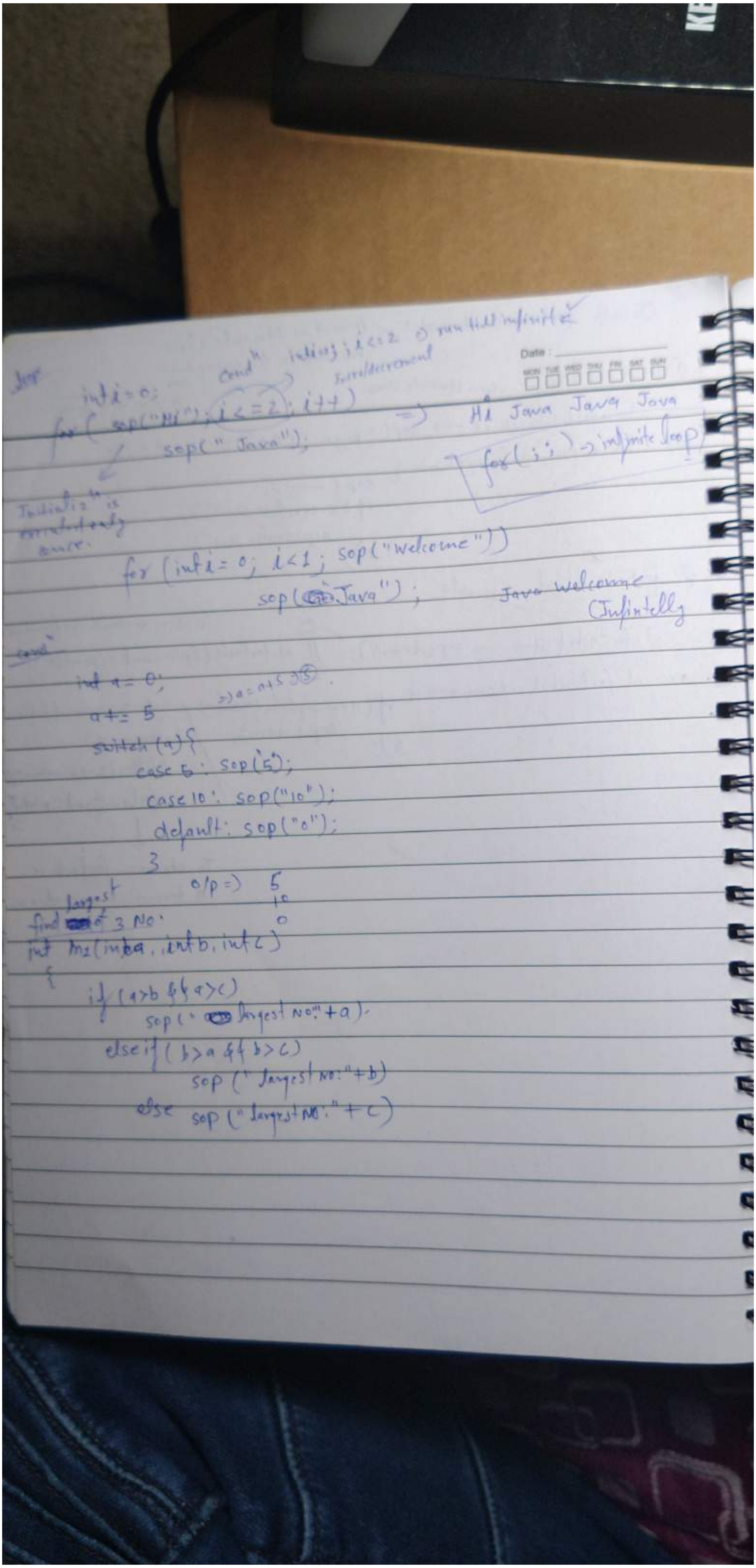
```
al.forEach( item → sop(item));   or  // al.forEach(System.out::println);   // Using method reference

or al.forEach( item → {
                 if(item.equals("[illegible]"))
                     sop(item);
              });

// using Stream & filter
al.stream()
    .filter(s → s.contain("[illegible]"))
    .forEach(Sys.out::println);
```

↓

In stream forEach is a terminal operation.



loop

condⁿ initiaz; i<=2 ⇒ run till infinite
incr/decrement

```
int i=0;
for ( sop("Hi"); i<=2; i++)
        sop(" Java");
```

⇒ Hi Java Java Java

Initializⁿ is executed only once.

for(; ;) → infinite loop

```
for (int i= 0; i<1; sop("welcome"))
        sop("Java");
```

Java welcome (Infintelly

condⁿ

```
int a= 0;
a+= 5          => a= a+5 ⇒ 5
switch (a){
    case 5: sop("5");
    case 10: sop("10");
    default: sop("0");
    }
```

o/p ⇒ 5
10
0

find largest of 3 No.

```
int m1(int a, int b, int c)
{
    if (a>b && a>c)
        sop(" largest No:"+a).
    elseif ( b>a && b>c)
            sop (" largest No:"+b)
    else sop (" largest No:"+c)
```

Java Programs

Rate yourself in a scale of 1 to 5. → whatever the area you are lagging

Date: MON TUE WED THU FRI SAT SUN □□□□□□□

- → ~~Inheritance prog/Dynamic polymorphism~~ → [illegible] last page
- → ~~String comparison by content & ref.~~ → [illegible] 2nd [illegible]. last page
- → ~~For 1 to 50 integer using for, while, do-while loop~~ NB2-Pg3 ⇒ JSE
  - ↓ ~~if no. contains 5 print the no.~~
- → S class const^r is called 1^st then subclass const^r when we create obj of subclass.

```
class P{
        P(){
            sop("P1");
        }
}
                                class C{
                                    C(){
                                        sop("C1");
                                    }
                                    psvm(S...)
                                    {
                                        C obj=new C();
                                        obj.show();
                                    }
                                    P.v show()
                                    { sop("S1");
                                    }
                                }
```

⇒
```
P1
C1
show
```

- → ~~Replace ; with , in string~~ ⇒ split & then String.join(",", array)
  - st.split(";") ↓
  - ~~String st = "1;2;3;4;5"; without using replace method.~~
- → WAP to print repeated char & its occurrance
  - "what a good day"
- → WAP to find total no. of occurrence of given char in a String without using any loop. (Java is Java again)
- → ~~How to remove duplicates from array list~~ (without using Set)
- → ~~To find the smallest & largest no in an array.~~
  - ↓ can be convert to list vs Collections.max(l); min
- → ~~Armstrong no.~~
- → find largest of 3 no.

Date : ____________

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

⇒

```
*
* *
* * *
* * * *
```

```
1
2 2
3 3 3
4 4 4 4
```

→ Reading Data from excel, properties file.

→ How to sort Arraylist (~~[illegible]~~ of obj) calln → Collections.sort(al); ⇒ ASC ord → returns void.

Collections ↓ A class Contains staticmethods.

sort Array → Arrays. sort (a) → array.

→ How to convert ~~an~~ Array into ~~[illegible]~~ list & vice versa.

List al = Arrays.asList (array) ↓ new String[] {"Hi", "Java"}

String S[]= al.toArray ( )

⇒ To reverse a list ⇒ ~~[illegible]~~ Collections.reverse(al); → returns void

How to find accurrance of word in a String. Hi Java, Hello Java → Java → .split(" ") convert to list → → .frequency (al, "xyz") } int

int m1 (S1, S2)

· max (al) → sort in ASC order last → max

· min (al) 1st → min

What are methods in ~~[illegible]~~ → Hashmap! { put(K,V), get(K), remove(K), putAll(hm) entrySet(), KeySet(), values () containsKey(), containsValue() clear(), isEmpty, size(), equals() }

L: add (K)
add (i, K)
get (i)
remove (i)
set (i, K) ⇒ Update

⇒ Program to ~~[illegible]~~ count No. of object created for a class.

BD - 1st Dec.

To count the pairs whose sum is a given number.

Date:
MON TUE WED THU FRI SAT SUN
□□□□□□□

(1) In an integer array, Count pairs with given Sum.
Print the pairs Count with given sum.

```
int [] a = {1, 2, 3, 4, 5, 6, -1};        {1,2,3,0,-1}
   int sum = 5;                           sum = 2,  (2,0),(3,-1)
                          (2,3),(1,4),(6,-1)
   get PairsCount (a, sum);

p.s.v getPairsCount (int[] a, int sum)
{
    int count = 0;

   // Consider all possible pairs & check their sums.
   for (int i = 0; i < a.length; i++)
      for (int j = i+1; j < a.length; j++)
           if (a[i] + a[j] == sum)
               count ++;

   sop(" Count of pair is: " + count);
}
```

⇒ 2 for loops ⇒
Time Complexity $O(n^2)$

(2) To count pairs whose sum is an odd no.
~~while loop~~

```
if ( (a[i]+a[j]) % 2  != 0 )  ⇒ for odd No.
                          == 0  ⇒ for even No.
    count ++
```

(3) WAP To find Intersection b/w Two arrays. (common ele)

```
for (int i = 0; i < a1.length; i++)
    for (int j = 0; j < a2.length; j++)
        if (a1[i] == a2[j])
            hashset.add(a1[i]);
```

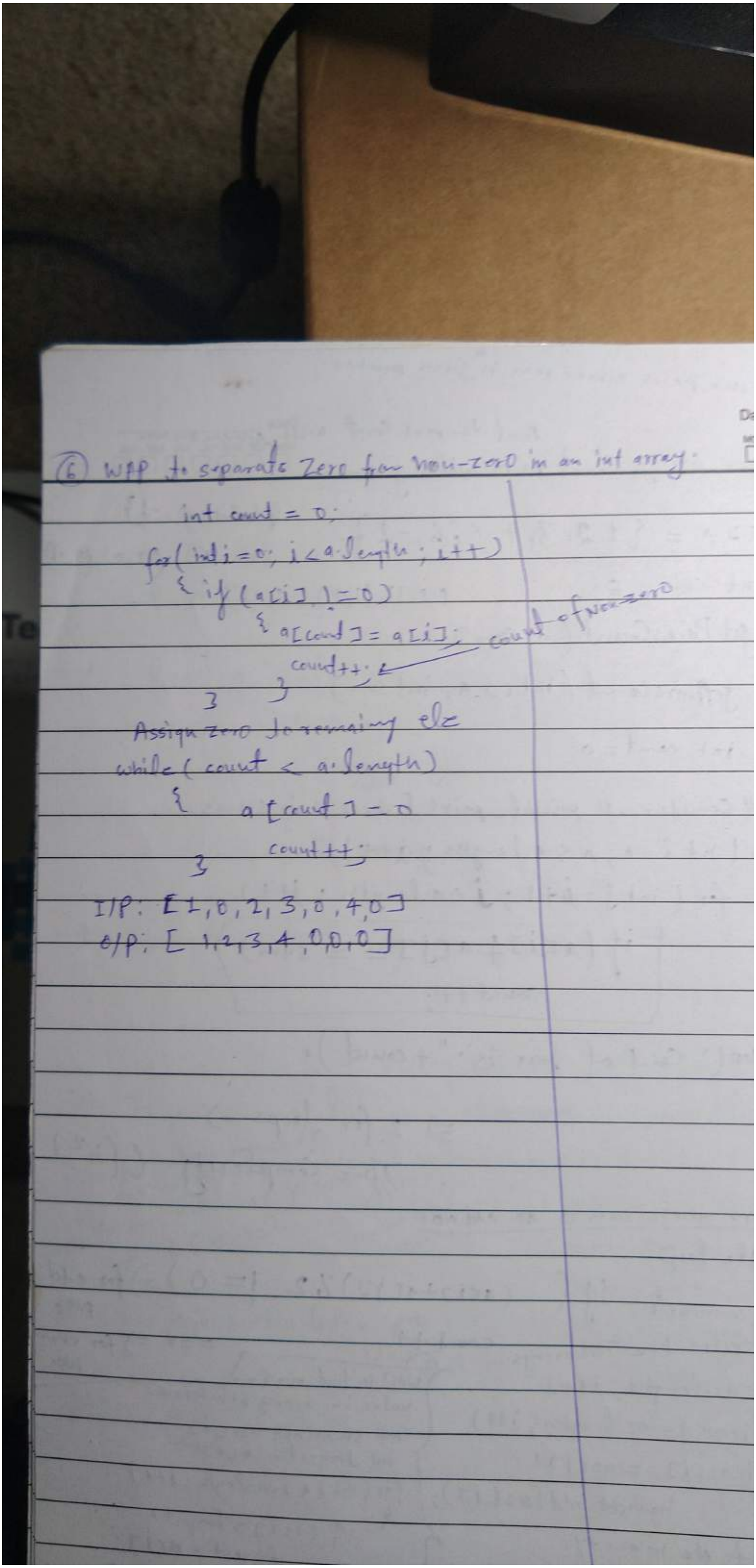
(4) WAP To find duplicate ele in array.

```
String s[] = {"x", "y", "x"};
HashSet<String> hs = new HS<S>();
for (String st : s)
    if (!hs.add(st)) { sop("duplicate ele is" + st); }
```

also can be solved as. find occ. of each char. in a string.

(5) WAP to find min & max value in array of integers.

```
int smallest = a[0];
int largest = a[0];
for (int i ; i < a.length; i++)
{   if (a[i] > largest)
        largest = a[i];
    elseif (a[i] < smallest)
        smallest = a[i]
}
sop(largest + " " + smallest);
```



(6) WAP to separate Zero from non-zero in an int array.

```
      int count = 0;
   for( int i=0; i<a.length; i++)
     { if (a[i] != 0)
          { a[count] = a[i];      count of non-zero
            count++;
          }
     }
  Assign zero to remaining ele
  while( count < a.length)
     {      a[count] = 0
            count++;
     }
```

I/P: [1,0,2,3,0,4,0]
O/P: [1,2,3,4,0,0,0]

→ Find first Non-repeated (Unique) character of a String.

→ check count of every char & store in LinkedHM
→ then loop map to find 1st char whose cnt is 1

```
P. S. char getFirstNonRepeatedChar(String s) {
    Map<C^r, I^r> hm = new LinkedHM<>(s.length());
                           ↓ maintains insertion order
for(char ch: s.toCharArray())
        hm.put(ch, hm.containsKey(ch) ? hm.get(ch)+1 : 1);
for( Entry<C^r,I^r> entry: hm.entrySet()){
        if ( entry.getValue() == 1){
                return entry.getKey();
        }   }
        throw new RuntimeExcep^n("didn't find any
                                 non-reptd char");
  }
```

How to count vowel Vowels & Consonants in a String.

```
  int count = 0        toLowerCase() Its a Alphabetic String
                                   No Digits, No Special char
  for ( char c : s.toCharArray()) {
        switch(c){
                  case 'a':
                   case 'e':
                   case 'i':
                    case 'o':
                    case 'u':
                            count ++;
                    break;
                    default:
              }
     }
    ~~sop("No. of vowels:" + count);~~
    sop.("No. of consonent" +(s.length() - count));
```

Trade off → com
b/w Th

Date : ___

MON TUE WED
□ □ □

Date : ____________

MON □ TUE □ WED □ THU □ FRI □ SAT □ SUN □

Locators
waits
webtables
actions class; Drag & Drop

```
list<WE> rows = d^r.fEs(By.xpath("//table/tbody/tr"));
for(int i=2; i<= rows.size(); i++ ){
        WE nameel = dr.fE(By.xpath("//table/tbody/tr[" +i+ "]/td[1]"));
    if (nameel.getText().toLowerCase().equals("mukesh")){
            WE checkbx = d^r.fE(By.xpath("//table/tbody/tr["+i+"]/td[3]"))
        }   checkbx.click();
efficient way }

② List<WE> Checkboxes =d^r.fEs(By.xpath("//table[@name='EmpTable']/tbody/tr/
                    td[contains(.,'"+name+"')]/following-sibling::td[2]"));
   for(WE e : checkboxes)
           e.click();   } or  checkboxes.forEach(e → e.click());
     export.click();

= //tr[.//*[starts-with(.,'"+text+"') and starts-with(@id, 'name-')]] //input=>
```

```
P.V. export(text)
{
    List<WE> cbox = d^r.fE(By.xpath(" "));
      // tr[.//*[starts-with(., '"+~~text~~"')]] //=>
          //input[@type='checkbox']

cbox.stream()
  .~~map~~(e → e.click())
  .forEach(e-> e.click())

    for (int i = 0; i <= cbox.size(); i++)      // To select only
        {                                          even checkb
            cbox.get(i).click();                   if (i%2==0)
        }                                          odd
            export.click();                          if (i%2!=0)
}
```

Selenium Que.

Date: ____________

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

→ implement Soft Assert, without using SoftAssert class

→ How wud you write your custom method to wait for a loading indicator or spinner to disappear.

→ To check whether an ele/name/~~text~~ is present in the webtable

5, 7, 10

3	4	5
6	7	8
9	10	11

(Jpmc)

→ How to handle stale element ReferenceExcep^n

clarivate

→ Diff b/w "/" // in xpath

/ for Direct child (inside Tag)

// for descendents

following is diff^t from descendents.

Artists, find

→ following vs following-sibling.

d.f(By.xpath(String.format(x, name))

String x = "//tr[.//*[contains(., '%s')]]//button[contains(., 'Edit')]

→ (1)

Name	Address	Action	status
Albert D	xyz	Edit	Active
austin V	xyz2	edit	Active
Rae A	xyz3	edit	Active
carlos B	xyz7	edit	Active

Click on edit for a given username)

(2)

→

@id = name-0

@id = name-1

Name	Address	Action
Ram V		□
Ramesh D		□
Raja Ram		□

export

~~Select~~

select all even or odd checkboxes first

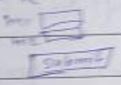
→ we can use two lists as well. & keep

select only those checkboxes ~~where~~ for username = 1^st 3 letter is Ram.

ele.getAttribute ("innerHTML")

How to handle when ele.getText() is not working

How to handle when submit button.click() is not working

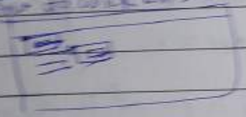


ele.sendKeys(Keys.TAB)

ele.sendKeys(Keys.Enter)

Actions class

→ How to click on submenu items which appears on mouse hovering in main menu



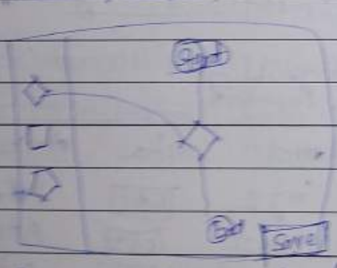
→ create work flow. Drag & Drop

→ ~~Store~~ store the values in list & return it from the method

9	8	3
1	5	6
4	7	2

drag & Drop

→



→ How to parameterize locator for a given text. 'x'

String ele = "//*[contains(text(),'%s')]"

d.fE(By.xpath(String.format(ele,x)));

d.fE(By.xpath("//*[contains(text(),'"+x+"')]"));

How to verify whether a table has the given column.

JavaScriptExecutor → interface

→ js.executeScript("arguments[0].click();", element) → To click an element

→ js.es("arguments[0].scrollIntoView(true);", element) → To scroll to an ele.

→ 'alignTo' boolean value
→ optional
→ true → Top of visible area
→ false Bottom
→ if ommitted → Top

→ js.es("window.scrollBy(0,50)"); → vertical scroll down
0, -50 → scroll up
0, document.body.scrollHeight → To scroll to bottom of Page.

→ js.es("arguments[0].value = arguments[1]", ele, "Hi Java");

Date : ____

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

TestNG

→ Random alert pop-up handling in TestNG. @BeforeMethod ~~check if alert present~~

→ When landing page of your appⁿ is loaded within 10 sec an alert or iframe pop appears on the page which block your operation on the page. How would you handle this.

can be added in @B M. if not sure when appears if using TestNG. ⇒ wait.until(ECʰ.frameToBeAvailableAndSwitchToIt) 'IsAlertPresent())

iframe{ Alert }

```
try {
    d. switchTo() frame("Id");
       close.click()
    d. switchTo().alert().accept()
}
Catch(NoSuchFrameEx e)
    Next.
{
}
```

→ How to run a T.C. 10 times → invocationCount.

→ Diff b/w @Beforesuite @BeforeTest & @BeforeMethod

→ How do you run failed Test Cases.

When click is not working on any ele due to some issue. (even after applying implicit & explicit wait. wait Until Clickable()

⇓ We can use JavaScript Executor → when all attempts failed ⇒ (also used to perform action which by Selⁿ cannot perform like scroll up/down)

→ We can click an Invisible ele which is present in the dom by forcefully making it visible using JSE.

Ways to click

→ ele.click()

→ jse.eS("arguments[0].click()", ele) // JSE

→ actions.moveToElement(ele).click().perform().

Awareness of concepts & ability to implement it

Able to write code

→ Design related que. to SSE & TL

Date : ______

MON □ TUE □ WED □ THU □ FRI □ SAT □ SUN □

→ Framework Design: Hybrid, DataDriven, Page Object Model.

Why do you think f/w is suitable for your req?

→ Metrics Used to share with Clients

→ Best Automation Practices

→ Team Handling experience }
→ Individual Contributor }

→ Creation & Execution of Test Plan / Test Autom^{n} strategy

TestManagement/BugTracking Tools: ALM, JIRA, Rally.

WebTable

Book Name	Author	Subject	Price
Learn Selenium	Amit	Selenium	300
Learn Java	Mukesh	Java	3000
Learn JS	Animesh	Javascript	1000
Master In Selenium	Mukesh	Selenium	2000
Master In JS	Deepak	JS	1000

Date : ____________

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

starts with 1
e.g 1, 2, 3, ...

we call it predicate

1 Print All headers of a Table.

```
List<WE> headers = d.fEs(By.xpath("//table[@name='BookTable']/tbody/tr[1]/th"))
headers.forEach(header -> sop(header));
```

2. Retrieve & print no. of rows

```
List<WE> rows = d.fEs(By.xpath("//table/tbody/tr"));
sop( rows.size() ); -> Including headers
sop( rows.size()-1); excluding header (only data rows)
```

3. Retrieve name of book whoose author is Mukesh

```
List<WE> books = d.fEs(By.xpath("//table/tbody/tr/td[text()='Mukesh']/../td[1]")).
books.forEach(book -> sop(book));
```

/.. -> points to parent

or

```
List<WE> books = d.fEs(By.xpath("———— 'Mukesh']/preceding-sibling::td[1]"))
```

4. Retrieve book name for which price is 1000

```
List<WE> books = d.fEs(By.xpath("//table/tbody/tr/td[text()='1000']/preceding-
                                sibling::td[3]"))
```

we can also give td[4] here

3 <td>
2 <td>
1 <td>
ref. -> <td>

or if td has some attribute

```
sibling::td[contains(@name, 'bookName')]
```

5. List all books where price >= 1000

```
~~WebTab~~ WE mytable = d.fE(By.xpath("//table[@name='bookTable']/tbody")).
List<WE> rows = d.fEs(By.tagName("tr")); // Not need, Rows can be found in one locator
for(int i=0; i< rows.size(); i++)
{
List<WE> cells = rows.get(i).findEs(By.tagName("td"));
if (Integer.parseInt(cells.get(3).getText())>= 1000) -> SOP(cells.get(0).getText
```

d -> myTable

```
for(WE row : rows) ->
List<WE> cells = row.fEs(..)
```

Date: ______________

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

6 print last row data

```
List<WE> cells = d.fEs(By.xpath("//table//tbody//tr[last()]//td"));
        cells.forEach(cell -> sop(cell.getText()));
```

7 print total cost of all books listed

```
  List<WE> costCells = d.fEs(By.xpath("//table//tbody//tr//td[4]"));
     int sum = 0;
     costCells.forEach(cost -> {   sum = sum + Integer.parseInt(cost.getText());
                                 });
     sop(sum);
```

8 Retrieve cell value for given col & row value.

```
        public static String getCellValue(int row, int col)
         {
            return d.fE(By.xpath("//table//tbody//tr[" + row + "]//td[" + col + "]")).
         }                                                                  .getText();
                                                  tr[2]//td[3]
```

9 Retrieve column Index using columnName   predicate

```
     public static int getColumnIndex(String colName){
        List<WE> headers = d.fEs(By.xpath("//table//tbody//tr//th"));
      int i = 0;
      headers.forEach( header -> {   if (header.getText().equalsIgnoreCase(colName))
                                          return(i+1);
                              else
                                  });  i++;
        for( int i=0; i<h.size();
                               i++)
        {
            if( headers.get(i).getText().equalsIgnoreCase(name))
                 return (i+1);
           }
           return null; };
```

10 print all data of table (works for dynamic tables as well where cells are not known)

```
  List<WE> rows = d.fEs(By.xpath("//table//tbody//tr"));
     for(WE row : rows) {   List<WE> cells = row.fEs(By.tagName("td"));
                          for(WE cell : cells)
                              sop(cell.getText());
                      }
```

others

Date : ____________

MON	TUE	WED	THU	FRI	SAT	SUN
□	□	□	□	□	□	□

what is your day to day activities in automation

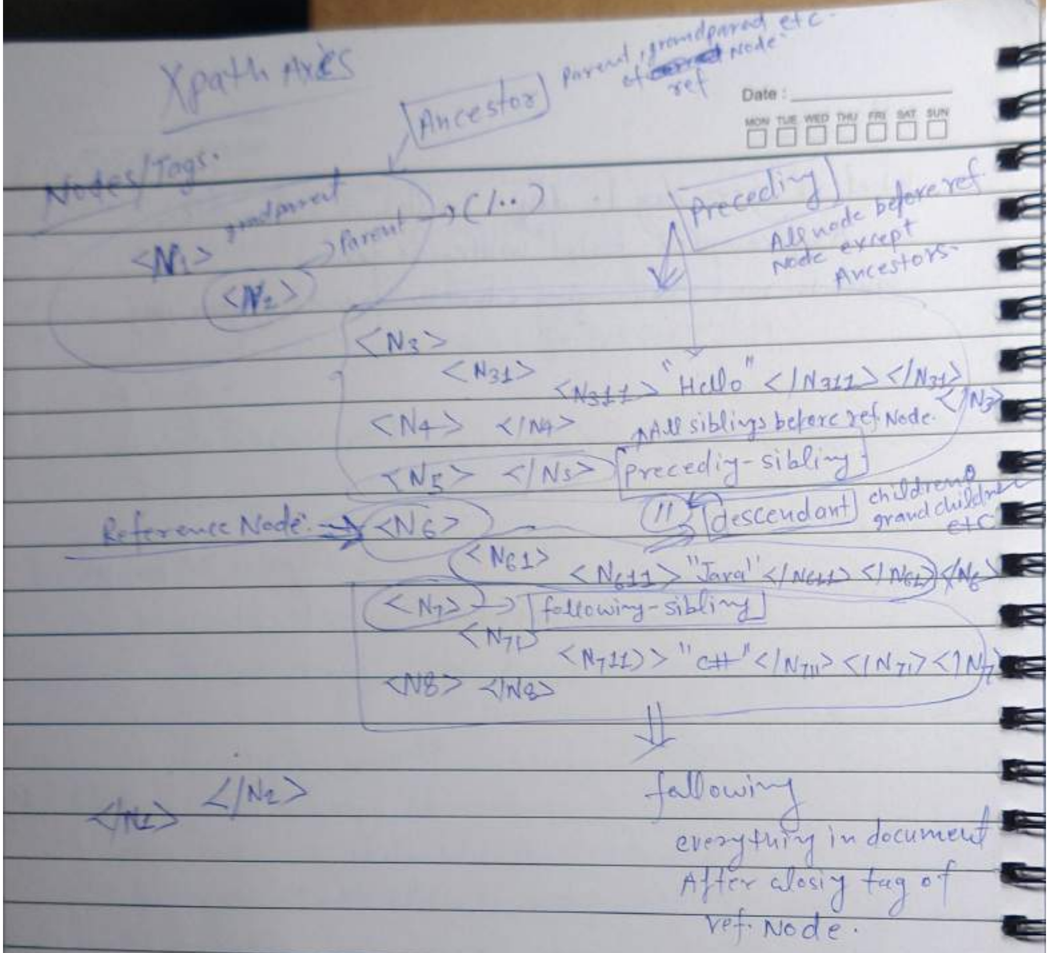
Any complex script written so far.

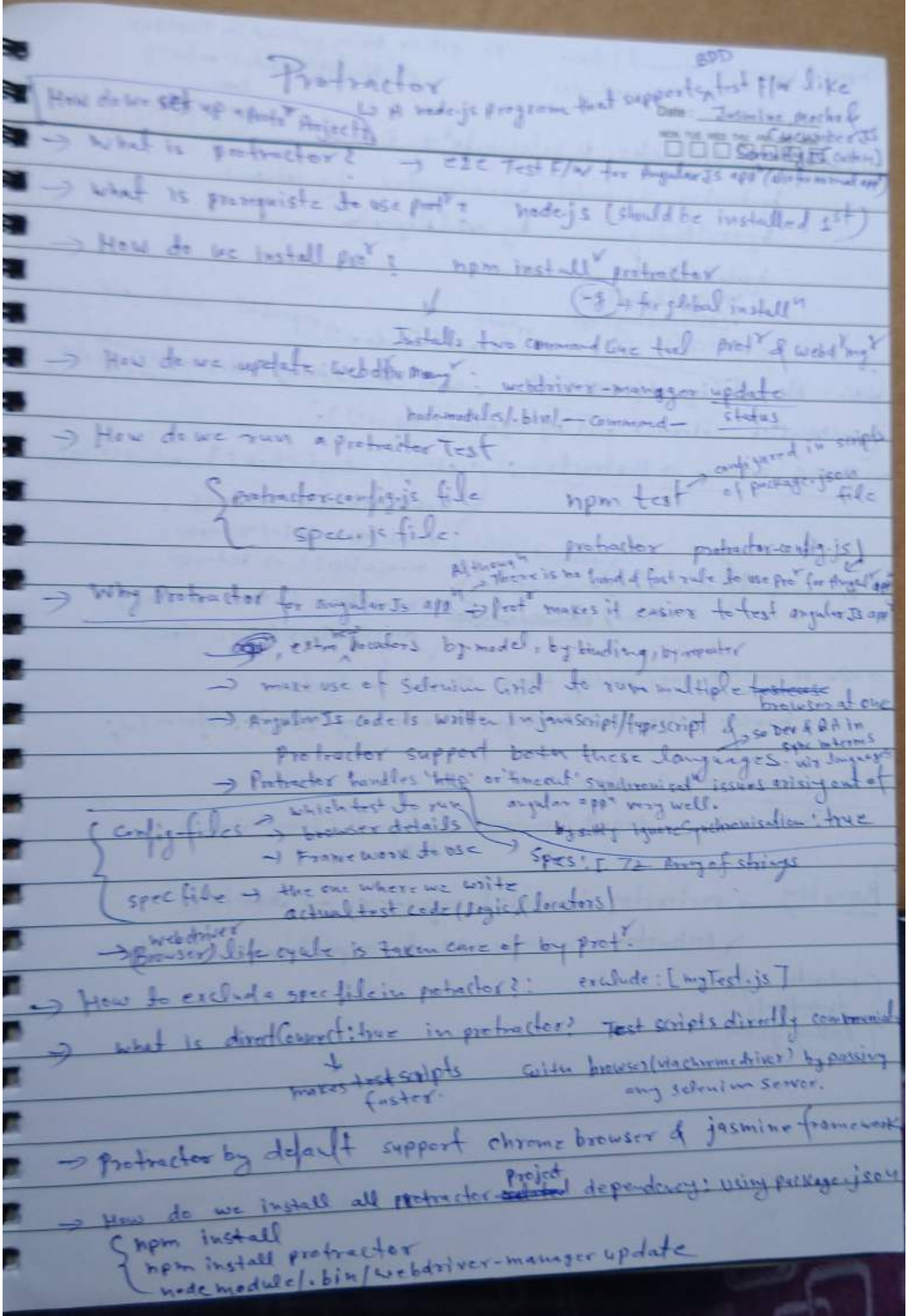
what is most challenging situation faced in your proj

Xpath Axes

Date : ___

MON □ TUE □ WED □ THU □ FRI □ SAT □ SUN □


Nodes/Tags.
Ancestor
Parent, grandpared etc. of ~~parent~~ Node ref
grandparent
parent
(/..)
<N1>
<N2>
Preceding
All node before ref Node except Ancestors.
<N3>
<N31>
<N311> "Hello" </N311> </N31>
</N3>
<N4> </N4>
All siblings before ref. Node.
<N5> </N5>
preceding-sibling
Reference Node.
<N6>
//
descendant
children, grand children etc
<N61>
<N611> "Java" </N611> </N61> </N6>
<N7>
following-sibling
<N71>
<N711> "C#" </N711> </N71> </N7>
<N8> </N8>
</N2>
</N1>
fallowing
everything in document After closing tag of ref. Node.



Protractor

How do we set up a prot^r Project?

↳ A node.js program that supports test f/w like BDD Jasmine, mocha & cucumber JS, Serenity JS (custom)

→ What is protractor? → e2e Test f/w for AngularJS app^n (also for normal app)

→ What is prerequiste to use prot^r? node.js (should be installed 1st)

→ How do we install pro^r? npm install^r protractor

(-g) ↳ for global install^n

↓

Installs two command line tool prot^r & webd^{r}mg^r

→ How do we update webdrivermang^r: webdriver-manager update / status

node-modules/.bin/ — command —

→ How do we run a protractor Test

{ protractor-config.js file
{ spec.js file.

npm test → configured in scripts of package.json file

protractor protractor-config.js

→ Why Protractor for angularJs app^n → prot^r makes it easier to test angularJS app

Although^n there is no hard & fast rule to use pro^r for AngularJS app

→ ~~app~~, extra^{ng} locators by.model, by.binding, by.repeater

→ make use of Selenium Grid to run multiple ~~testcases~~ browsers at one

→ AngularJS code is written in javascript/typescript & Protractor support both these languages. → so Dev & QA in sync in terms wrt languages

→ Protractor handles 'http' or 'timeout' synchronisat^n issues arising out of angular app^n very well.

by setting ignoreSynchronisation: true

{ config files → which test to run → Spes: [ ] → Array of strings
→ browser details
→ Framework to use

spec file → the one where we write actual test code (logic & locators)

→ webdriver (Browser) life cycle is taken care of by prot^r.

→ How to exclude a spec file in protractor?: exclude: [myTest.js]

→ What is directConnect: true in protractor? Test scripts directly communicate with browser (via chromedriver) bypassing any selenium server.

↓
makes ~~test~~ scripts faster.

→ Protractor by default support chrome browser & jasmine framework

→ How do we install all ~~protractor~~ project ~~related~~ dependency: using package.json

{ npm install
npm install protractor
node module/.bin/webdriver-manager update

import { browser, by, element } from 'protractor';

To get an value entered in ~~Textbox~~
ele.getAttribute("value")

Date: ____________

MON □ TUE □ WED □ THU □ FRI □ SAT □ SUN □

→ Locators in protractor: & selenium loc + Below

by.binding, by.model, by.repeater, by.options
by.exactBinding, by.exactRepeater
by.buttonText
· partialButtonText
· cssContainingText
· deepCss

{ addBtn = element(by.id('addBtn')) → returns ElementFinder → represent single ele.
{ selectOptionList = element.all(by.tagName('select')). → returns ElementArrayFinder → represents array of ele.

→ mouse actions in Protractor: { click(), doubleClick(), contextClick()
mouseDown(ele) → press
mouseMove(ele) → move
mouseUp() → release
mouseMove(x-offset, y-offset)
dragAndDrop(ele1, ele2)

→ browser.wait(ExpectedCond.visibilityOf(ele), 4000)
presenceOf(ele)

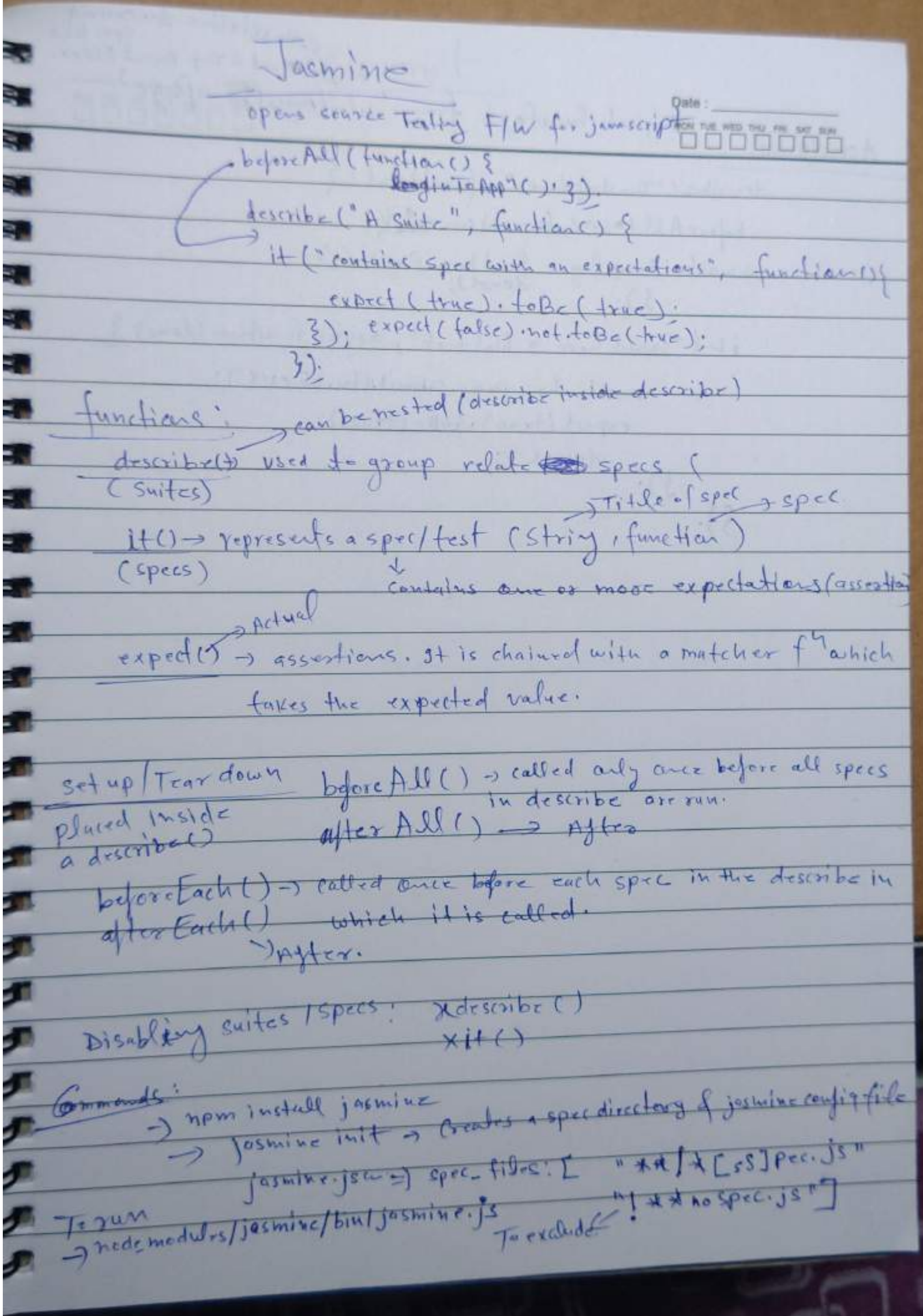
→ How do you stop page loading when element is loaded
browser.actions().sendKeys(protractor.Key.ESC).perform();

→ Assertion in Protractor

Jasmine → expect(something1).toEqual(something2)
· toBe

→ Reporting { protractor-beautiful-reporter → Jasmine
{ protractor-cucumber-html-reporter → Cucumber

{ protractor.config.js
ts-config.ts
package.json



Jasmine

opens source Testing F/W for javascript

Date : ____________
MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

```
beforeAll ( function() {
        loginToApp"( ); });
describe("A Suite", function() {
    it ("contains spec with an expectations", function(){
        expect ( true). toBe ( true);
    }); expect( false).not.toBe(true);
});
```

functions:

describe() (Suites) → can be nested (describe inside describe) used to group related specs (

it() (specs) → represents a spec/test (String → Title of spec, function → spec)
↓ contains one or more expectations (assertio

expect() → Actual → assertions. It is chained with a matcher fⁿ which takes the expected value.

set up / Tear down placed inside a describe()

beforeAll() → called only once before all specs in describe are run.
afterAll() → After

beforeEach() → called once before each spec in the describe in which it is called.
afterEach() → After.

Disabling suites / specs: xdescribe()
xit()

Commands:
→ npm install jasmine
→ Jasmine init → Creates a spec directory & jasmine config file
jasmine.json ⇒ spec_files: ["**/*[sS]pec.js"
To exclude → "!**no Spec.js"]

To run
→ node_modules/jasmine/bin/jasmine.js

ॐ

Date :

What is Default time out for a protractor with Jasmine spec to fail

by default 30 sec.

To change it jasmineNodeOpts: {defaultTimeoutInterval: in millisec 60000}

1 min

→ Jasmine has built in assertions but mocha doesn't have assertions library & uses Chai for assertions.

Path relative to current Spec file.

~~#~~ ../ → To go Parent folder.

Date : ________

MON TUE WED THU FRI SAT SUN

□ □ □ □ □ □ □

Async/Await

```
            import homPage from '../../pages/homePage')

    describe("The dropdown", function() {
     beforeAll(async function(done){
    await homePage. loadHomePage();
                    done();
            });
     it("should have a hint text", async function(done) {
            await homepage.submitBtn.click();
             expect(true).toBe(true);
               done();
          });
       });
```

JDBC (Java DataBase Connectivity)

→ latest JDBC version 4.3

Date: ____________

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

→ It's a Java API to connect & execute SQL query with database
↓ Select, Insert, Delete, update etc.

→ There are 4 types of JDBC Dr
we use Type 4 Dr (Thin Dr) as it has better performance than all others.

⑤ steps -

→ Reg Dr class is optional since JDBC 4.0

① Class.forName("~~[illegible]~~ com.microsoft.sqlserver.jdbc.SQLServerDriver");

② try (Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost:1433", "username", "password")) { → class
{

③ PreparedStatement ps = con.prepareStatement("select * from emp where id=?"); → parameter index
ps.setInt(1, 425); pre-compiled only once so faster than statement

④ ResultSet rs = ~~@~~ ps.executeQuery(); .executeUpdate()
while (rs.next()) previous(), first(), last()
{
sopl rs.getInt(1)); // rs.getInt("EmpId") → Datatype of col → By col label or index 1,2,3
sop(rs.getString("name"));
} }

⑤ close connection: since Java 7 JDBC has ability to use try-with-resources statement to automaticall close resources.

(con.close()) catch(SQLExcep ex) { }
catch(Excep ex) { }

Statement st = con.createStatement();
ResultSet rs = st.executeQuery("select * from emp");
int i = st.executeUpdate(—); ↳ returns int
ResultSetMetaData rsmd = rs.getMetaData(); → used to get details like col Name, Datatype length etc.

CallableStatement cs = con.prepareCall("{call myprocedure(?,?)}"); → to call stored proc & fns. proc
"{? = call sum(?,?)}"

Performance:

Statement < PreparedStatement < CallableStatement
↓ query is compiled everytime; ↓ compil" only once; → used for parameterized query

Date : ____________

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

rep. single unit of work

Transaction Mgmt in JDBC: → guided by ACID Prop. of txn

Atomicity → all sucessful or none

Consistency → only D.B. from one consistent state to another Cosst.

Isol": ensures one txn is isolated from other txn. (locking technique)

Durability - once txn commited change is permanent even in event of errors, power less etc.

Adv. of Txn M:- fast Performance coz D.B. is hit at the time of commit.

Connection interface has fallowing meth for txn mgmt.

```
void con.setAutoCommit(false); → By default set to true.
                         ↓
                       Boolean
void con.commit(); → commits txn
void con.rollback(); → cancels txn.

        con.setAutoCommit(false);
        Statement st = con.createStatement();
            st.executeUpdate("insert into emp values(1,'Abhi',4000)");
            st.executeUpdate(____);
            con.commit();
            con.close();
```

→ PreparedStatement: helps us preventing SQL Injection attacks coz it automatically escapes the special charactres.

→ allows us to execute Dynamic query with paramr inputs.

→ provides Setter meth to set the input pr for query

⇒ "Java.sql" pkg contains interfaces & classes of JDBC API

Classes:- DriverManager, Blob, Clob, Types.

Interfaces → Connection, statement, P.S., C.S, ResultSet, RSMD,

Singleton Design Pattern.

It restricts the instantiatⁿ of a class outside class & ensures that only one instance of class exists in the JVM.

Date : ________

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

⇒ Used for logging, driver objects, caching, thread pool.
also used in oth design patterns like Abstract Factory, Facade etc
e.g. core java classes, "java.lang.Runtime"
singleton classes:

```
public class A{
    private static final A obj = new A();
        private A(){
                  }        // private constructor to avoid class
                              instantiatⁿ outside class.
    public static getInstance(){
                return obj;
    }     }
```

A x = A.getInstance()

↓ singleton Pattern Implementⁿ has always been a controversial topic among devrs.

By Using Enums

```
public enum A{
        INSTANCE;
    }
```

enum fields are compiled time constants but they are instances of their enum type.

Internally
class A{
p.s.f. A INSTANCE = new A();
" A XYZ = new A()

A.INSTANCE

Most of class related implementⁿ of singleton patten are not full-proof in all cases.
like we can bypass private constructor using Serializⁿ or reflection concepts in java & we can create multiple instances of above class
⇓ so we can use enum to create singleton class.

→ cloning concept can also break singleton Pattern.

enum in Java

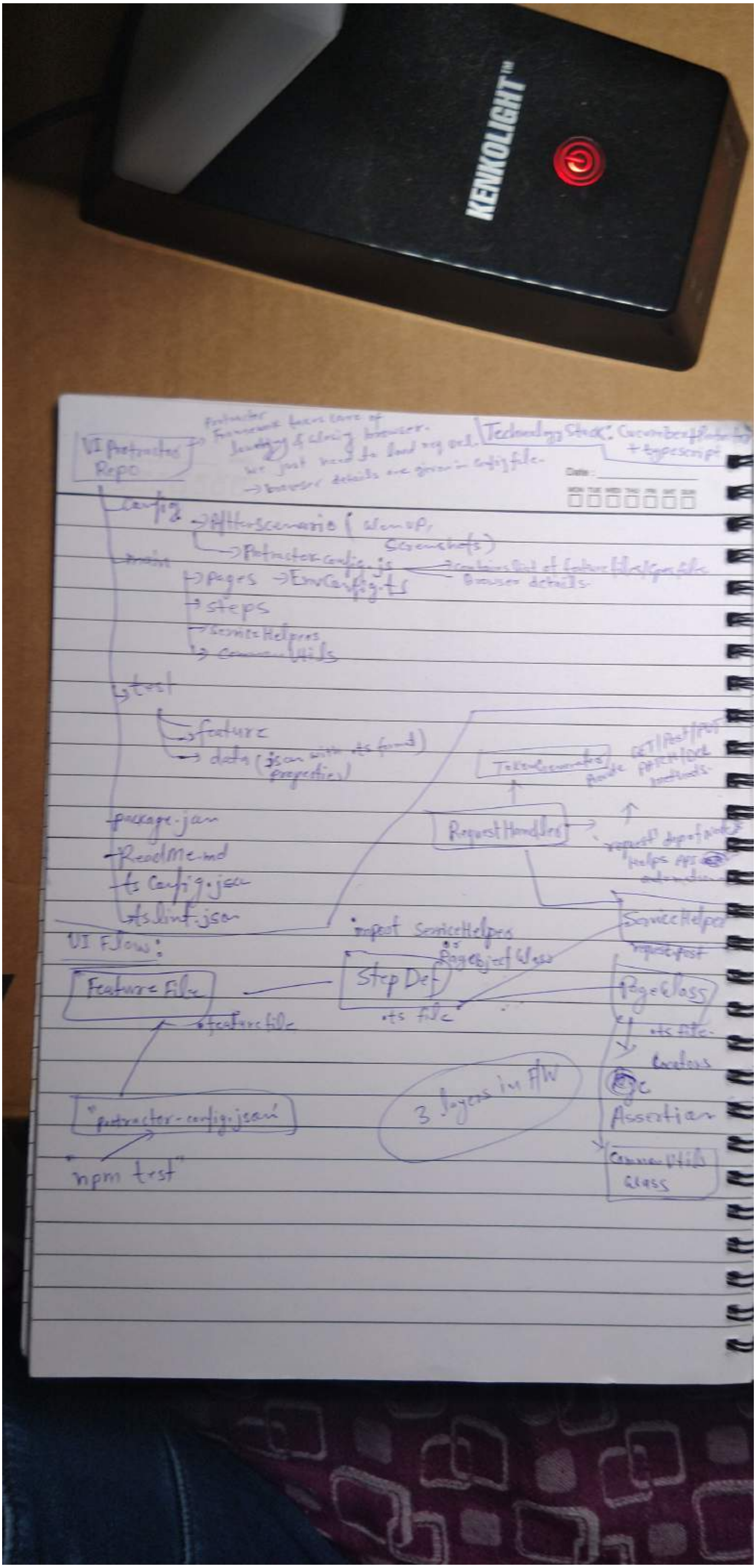
→ represents a group of named constants.
→ looks like classes & can have variables / methods & constructors.
only concrete meth
can hv main meth.
→ can be declared outside or inside a class but not inside any method.
→ First line inside enum should be list of constants. (all const in capl letter)
→ cannot extends anything (implicitly exends Enum class)

```
public enum Color
{
  RED,GREEN,BLUE;
  private Color()
    {
      sop(this.toString());
    }
  P. V. colorInfo()
    {  sop("universal color");
    }
}
```

```
p. class Test{
    p.s.v m(s...)
    { Color c1 = Color.RED;
       sop(c1);
       c1.colorInfo();
```

constr is called for each constant

```
O/P: RED
     GREEN
     BLUE
     RED
     Universal color.
```



UI Protractor Repo → Protractor Framework takes care of loading & closing browser.
We just need to load req url.
→ browser details are given in config file.

Technology Stack: Cucumber+Protractor + typescript

Date:
MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

config → AfterScenario (cleanup, Screenshots)
→ Protractor-config.js ← contains list of feature files/spec file, Browser details.

main
→ pages → EnvConfig.ts
→ steps
→ ServiceHelpers
→ CommonUtils

test
→ feature
→ data (json with ts format properties)

package.json
ReadMe.md
tsConfig.json
tslint.json

TokenGenerator

provide GET/Post/Put/Patch/Del methods.

RequestHandler

"request" dependency helps API automation

ServiceHelper

import/post

UI Flow:

Feature File

.feature file

Step Def

.ts file

import ServiceHelpers or PageObject Class

PageClass

.ts file

Locators

etc

Assertion

CommonUtils class

"protractor-config.json"

"npm test"

3 layers in F/W

Jbehave-Serenity
UI Repo

Tech stack: Jbehave+Serenity 1.4.x..2

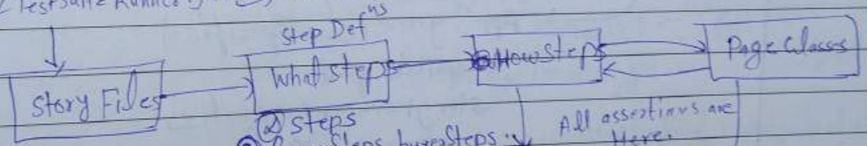
Date :

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

```
- src/main/java
      -> ~~whatsteps~~
      - How steps
      - Page classes
      - Utils
      - ~~Readers files~~
- src/main/resources
      -> Serenity.propeoties -> contains all serenity related config^n.
      -> Application.propeoties -> Env & testdata.
      - storyrunner.properties -> list of storyfiles.
- src/test/Java
      -> AccesptanceTestSuiteRunner.java   (reads)
-> src/test/resources
      - stories
- pom.xml
- ~~targets~~ -> site/serenity/index.html
```

4 Layers in F/W

A/L TestSuite Runner.java -> Runas JUnit



All assertions are Here.

```
@Steps
BuyerSteps buyerSteps;

Given User login App^n

@Given("user login app")
p.v. ~~login()~~
{
  buyerSteps.login();
}
```

```
public class BuyerSteps{
HomePage homePage;   <- Automatic instanciat^n & obj injection by Serenity
@Step -> To show meth^n in Serenity report
P.V. login()
{
  homePage.login();
}
}
```

```
P. class HomePage extends PageObject
{
  @FindBy(css="-").
  WebElement loginBtn;
  @FindBy (xpath="-")
  List<WE> textList;
  P.V. login(){}.
}
```

service
Maven Project

Service RepO

Tech stack: Jbehave + Spring Framework + Java

Date : ______

MON TUE WED THU FRI SAT SUN
□ □ □ ☑ □ □ □

src/main/java

Step Defs
Requst Pojo → lombok dependency → just declare variables it will provide you getters & setters methods
ResponsPojo

@Data
① JsonIgnoreProperties (ignoreUnknown = true)
② JsonInclude (NonDefault)

TokenGenr

Rest Config
Steps Config
enums

Utils
- cleanUP
- jsonFileReader
- JsonComposer
- json schema Validator
- PropertyManager
- YmlReader

src/main/resources
→ App^n.yml (env specific)
endpoints.yml
storyrunner.properties
messages.yml

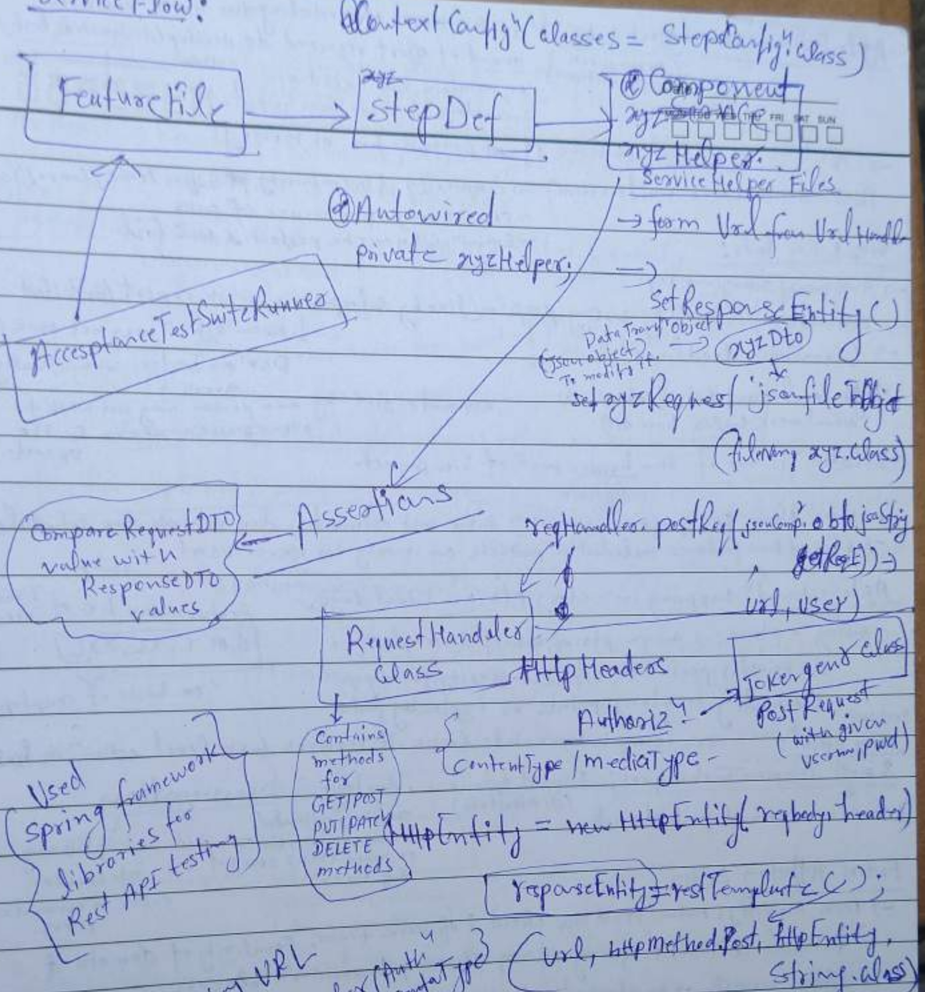
src/Test/Java
→ A/CTestSuiteRunner → reads storyrunner (featurerunner) file to get list of feature/stories files
rightclick → Run As Junit Runner.

src/test/resources
→ features
→ testdata
- json
- xml.

→ pom.xml
→ ReadMe.md

targets
→ reports
reports.html
(Jbehave Report)

Service Flow:


@ContextConfig^n(classes = StepConfig^n.class)
FeatureFile
xyz StepDef
@Component
xyzService
xyzHelper.
@Autowired
private xyzHelper;
AcceptanceTestSuiteRunner
Service Helper Files
→ form Url from UrlHandler
→
setResponseEntity()
Data Transf^r Object
(jsonobject) To modify it.
xyzDto
set xyzRequest(jsonfileToObject
(filename, xyz.class)
Assertions
Compare RequestDTO value with ResponseDTO values
reqHandler.postReq(jsonComp.objtojsonString(getReqE)) →
, url, user)
RequestHandler Class
HttpHeaders
Authoriz^n:
TokengenerClass
Post Request (with given username, pwd)
Contains methods for GET/POST PUT/PATCH DELETE methods
ContentType / mediaType -
HttpEntity = new HttpEntity(reqbody, header)
responseEntity = restTemplate();
(url, HttpMethod.Post, HttpEntity, String.class)
Used spring framework libraries for Rest API testing

→ forming URL

→ setting up header (Auth^n contentType)

→ modify a req body if need

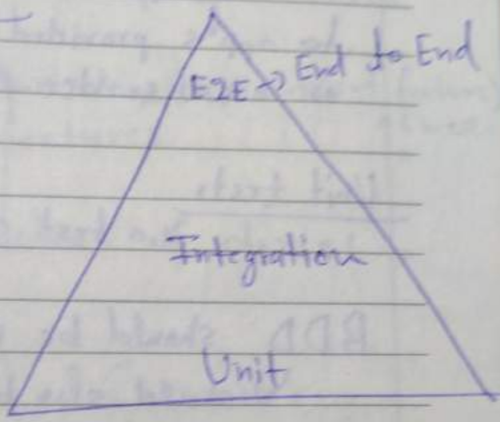
→ Hit the API

Where should you use BDD

Agile Test Pyramid

Less E2E test

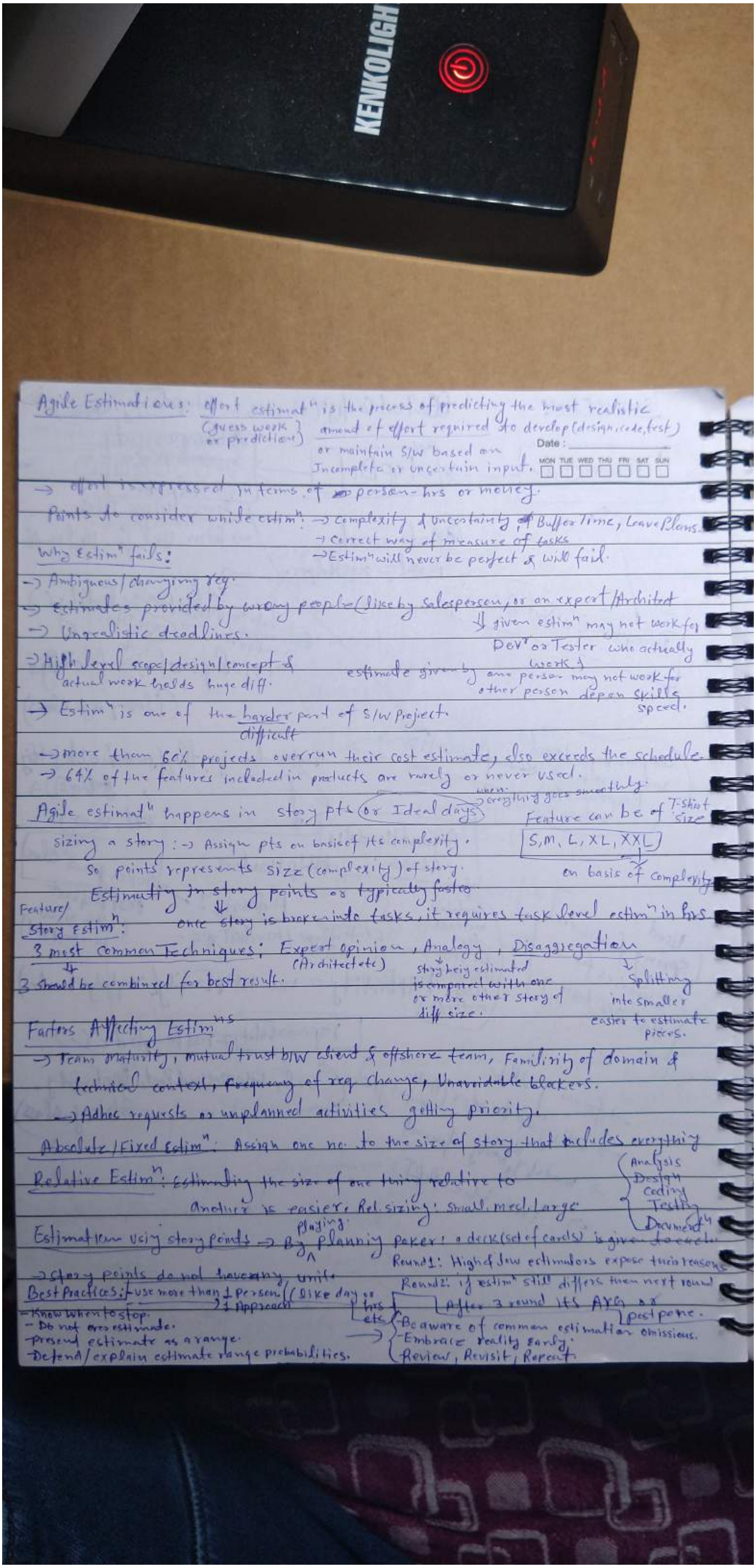
more Unit tests



No. of tests that are appropriate at diff^{t} level.

E2E: e2e tests checks your entire app^{n} it uses the same UI as users & verifies that someting actually works, this includes saving things in D.B.

↳ Use few e2e tests that verifies most impo^{t} parts, but don't use them for every thing.



Agile Estimations: effort estimatⁿ (guess work or prediction) is the process of predicting the most realistic amount of effort required to develop (design, code, test) or maintain S/W based on Incomplete or uncertain input.

Date : ____

MON □ TUE □ WED □ THU □ FRI □ SAT □ SUN □

→ effort is expressed in terms of person-hrs or money.

Points to consider while estimⁿ: → complexity & Uncertainty, Buffer Time, Leave Plans.
→ correct way of measure of tasks
→ Estimⁿ will never be perfect & will fail.

Why Estimⁿ fails:

→ Ambiguous / changing req.

→ estimates provided by wrong people (like by Salesperson, or an expert / Architect ⇓ given estimⁿ may not work for Devʳ or Tester who actually work.)

→ Unrealistic deadlines.

→ High level scope / design / concept & actual work holds huge diff. estimate given by one person may not work for other person depen skills speed.

→ Estimⁿ is one of the harder (difficult) part of S/W Project.

→ more than 60% projects overrun their cost estimate, also exceeds the schedule.
→ 64% of the features included in products are rarely or never used.

Agile estimatⁿ happens in story pts (or Ideal days) when everything goes smoothly.

Feature can be of T-Shirt size
S, M, L, XL, XXL
on basis of complexity

sizing a story :→ Assign pts on basis of its complexity.
So points represents size (complexity) of story.
Estimating in story points or typically faster

Feature / Story Estimⁿ: ⇓ once story is broken into tasks, it requires task level estimⁿ in hrs.

3 most common Techniques: Expert opinion (Architect etc), Analogy, Disaggregation
⇓ 3 should be combined for best result.
Analogy ↓ story being estimated is compared with one or more other story of diff size.
Disaggregation ↓ Splitting into smaller easier to estimate pieces.

Factors Affecting Estimⁿˢ

→ Team maturity, mutual trust b/w client & offshore team, Familirity of domain & technical context, Frequency of req change, Unavoidable blockers.
→ Adhoc requests or unplanned activities getting priority.

Absolute / Fixed Estimⁿ: Assign one no. to the size of story that includes everything { Analysis, Design, Coding, Testing, Documentⁿ }

Relative Estimⁿ: Estimating the size of one thing relative to another is easier. Rel. sizing: small, med, Large

Estimation using story points → By playing Planning Poker: a deck (set of cards) is given to each.
Round1: High & low estimators expose their reasons
Round2: if estimⁿ still differs then next round
After 3 round its AVG or postpone.

→ Story points do not have any unit.

Best Practices: - use more than 1 person / 1 Approach (like day or hrs etc)
- Know when to stop.
- Do not over estimate.
- Present estimate as a range.
- Defend / explain estimate range probabilities.
→ { - Be aware of common estimation omissions.
- Embrace reality Early.
- Review, Revisit, Repeat.

Agile → It is a philosophy for ways of doing things

Agile manifesto: (defn) → value { working S/W more over comprehensive Docn
Respondig to change over following a Plan.
Individuals & Interactions over Process & tools

Date: MON TUE WED THU FRI SAT SUN

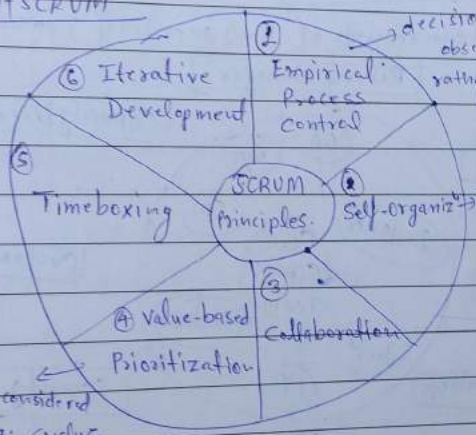
methods: → Kanban, Extreme Programy, Lean Development
→ SCRUM is most popular

Agile Benefits: { Quicker ROI
Lower Total cost ⇒ maximise value & quality
Respond to change
Reduce Risk

SCRUM → Name taken from sport of Rugby where everyone in the team acts together to move the ball down the field

Its a framework/Agile process that allows us on delivering the highest business value in the shortest time.

Key Principles of SCRUM
↓ core principle.



→ decisions are made based on observations & experimentation (experience) rather than on detailed upfront Planning.

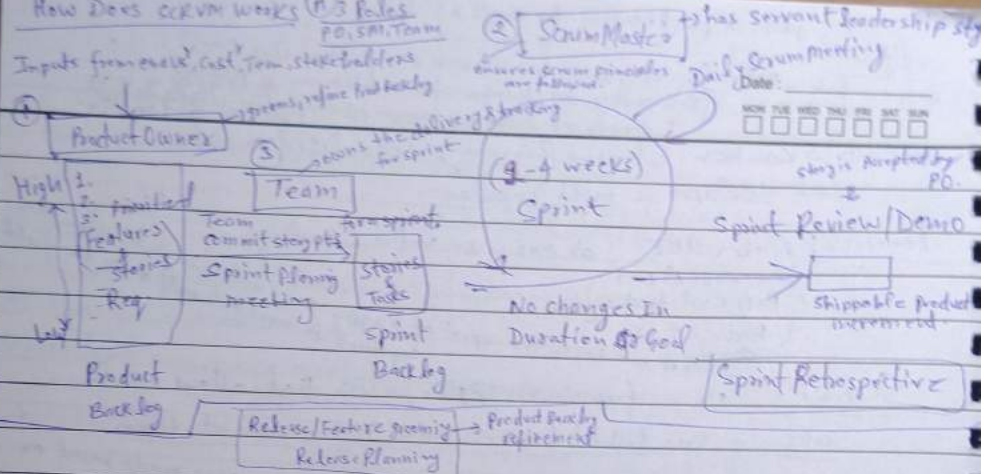
Self-organizn → everyone knows one's roles & resp & acts accordingly
PO, SMr, Dr, Tr
⇓
Needs to ensure Scrum principles are followed.

3 factors to be considered while prioritizing: { value
Risk or Uncertainity
Dependencies

Scrum Values: Commitment, Focus, Openness, Respect, Courage

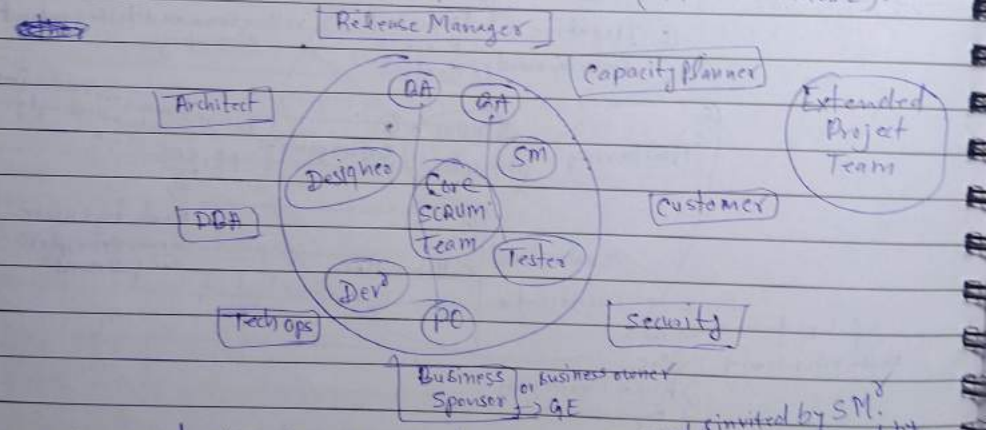
Where SCRUM is NOT Effective	is Effective
Lack of customer Involvement	Cr Involvement
Big sized team (> 8-10)	5-7
When budget & schedule is fixed.	Control & mangement is empirical (practical)
Hard to implement Scrum when team mbrs are not co-located.	work delivered is Incremental

How Does SCRUM works (A) 3 Roles PO, SM, Team



Inputs from execs, cust, Team, stakeholders
(1) Product Owner
→ grooms, refine Prod Backlog
(2) ScrumMaster
→ has servant leadership sty
ensures scrum principles are followed
Daily Scrum meeting
(3) Team
→ owns the delivery & tracking for sprint
High
1.
2.
3.
Prioritized
Features
Stories
Req
Low
Product Backlog
Team commit story pts for sprint
Sprint Planning meeting
Stories & Tasks
Sprint Backlog
(2-4 weeks) Sprint
No changes In Duration & Goal
story is Accepted by PO.
Sprint Review/Demo
Shippable product increment
Sprint Retrospective
Release/Feature grooming → Product Backlog Refinement
Release Planning

→ The core Project team consists of 5-9 members (7 plus or minus 2).



→ management style is decentralized. → organized & invited by SM.

(B) SCRUM Ceremonies/Rituals: (Release Planning), (1) sprint Planning, (2) Daily standup, (3) Sprint review, (4) sprint retrospective.

Sprint Rituals:

Product Backlog Refinement → Attended by PO, SM, Team (15 mins) SCRUM

(1hr) ↓ PO is final decision Maker on Acceptance of story

→ what worked well
→ what did not go well.
→ Action items for improvement
→ New Ideas
→ Appreciations

(C) SCRUM Artifacts:
- Product Backlog
- Sprint Backlog
- Burndown chart
- SCRUM Board, Points, velocity

⇒ Epic is broken into features & a feature is broken into stories.

SM: Needs to protect the team from over-commit. (due pressure from aggressive PO)
shields the team from any external interferences

⇒ Product Planning & Release planning are not considered part of Scrum Framework.

Sprint Planning: → conducted at the start of sprint. PO prioritizes & select backlog for sprint. → Commit Story Pts
→ create sprint backlog (tasks) from product Backlog items (user stories/features) as per sprint capacity
→ estimate sprint backlog (tasks) in hrs. Use template of story tasks.

Daily scrum - { what I did yesterday?
what I will do today?
Any blockers [illegible] team need to know [illegible] }

Product Planning: Business Owner & PO defines the product vision

Date:

MON TUE WED THU FRI SAT SUN
□ □ □ □ □ □ □

Release Planning: 2 to 3 days activity
[illegible]: 3-4 iterations 6-8 sprints

product Roadmap
High level planning (milestones & dates)

Burn down chart: includes [illegible] of user stories (backlog items)

→ represents work left to do versus time.

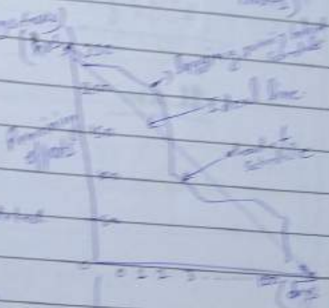
→ outstanding work (backlog) is often on vertical axis
with time along horizontal

→ It's a run chart of outstanding work.

→ useful in predicting when all of the work will be completed (per 'days')

→ based on task hrs (which has been assigned) move as completed on daily basis
graph is updated accordingly in tools like rally.

→ But [illegible] estimation is done in points at story level then
chart may not be updated on daily basis until end of sprint.



Velocity: [illegible] hrs/day
[illegible] hrs/sprint
[illegible] /sprint

Shows
{ work done on each iteration
work remaining
work done [illegible] for
when we can expect to [illegible] }

Types of userstories

Epic, business story, Tech debt, Spike story (R&D)

Infrastructure story (Env Setup), Document story.

Story has
&

Description	As a —	who
	I want to —	what
	So that —	why

AC:

Testing Strategy

Priority

Dependency

Definition of Done

Diff people for diff perspectives. [illegible] done is when code is complete
Tester — until integr testing is complete
customer → Install & deployment of system together
with user documentation

In SCRUM, "done" is an agreement b/w the PO & Dev Team.

[Demo] → DOD for story

SCRUM Tools: Licensed: Rally, JIRA, TFS (Team Foundation Server), VersionOne, Mingle

openSource: Ice scrum, Agilefant