```java
*WaitConcept.java
package seleniumsessions;

public class WaitConcept {

    public static void main(String[] args) {

        //wait -- sync between script vs app
        //auto wait


        //1. static wait: Thread.sleep(10000) -- Java lib
        //e1 --> 2 secs --> total timeout : 10 secs
        //e1 -- 0 secs --> total timeout : 10 secs
        //e1 -- 15 secs --> total timeout : 10 secs

        //2. dynamic wait: total time
        //10 secs: 2 secs -->
        //10 secs : 15 secs
            //a. Implicitly wait
            //b. Explicit wait
                //b.1: WebDriverWait
                //b.2: FluentWait

```

Manage method returns options class.

Timeout returns timeout class.

Implicit wait also returns timeout class.

Implicit is overloaded-

```
.impl
implicitlyWait(Duration duration) : Timeouts - Timeouts
implicitlyWait(long time, TimeUnit unit) : Timeouts - Timeouts
getImplicitWaitTimeout() : Duration - Timeouts
```

Second method is deprecated.

```java
        WebDriver driver = new ChromeDriver();
        driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);
```

```java
package seleniumsessions;

import java.time.Duration;

import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class ImplicitlyWait {

    public static void main(String[] args) {

        WebDriver driver = new ChromeDriver();
        //driver.manage().timeouts().implicitlyWait(10, TimeUnit.SECONDS);//sel 3.x

        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));//sel 4.x

        driver.get("");
        //global wait: it will be applied for all the webelements
        //login page: 10 secs
        //e1 : 10 --> 2 : 2
        //e2 : 10 --> 5 : 5
        //e3 : 10 --> 4 : 4

        //product page: 15 secs
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(15));//sel 4.x
        //e4
        //e5
        //e6

        //login page: 15 secs
        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(10));//sel 4.x
        //e1 e2 e3: 10 secs
```

15.00

The latest wait value will be applied globally.

So first ten seconds, then we increased to 15 seconds, so latest value which is 15 seconds will be applied globally, unless you change it further.

Unnecessary waiting for all elements-

```java
        //e1: 10 secs: 4 secs:
        //e2: imp wait: 0
        //e3: imp wait :0
```

E2 and e3 dont need waits.

Slows down everything.

Dont use implicit.

One bad approach is nullify the implicit wait-

```
34        //e1: 10 secs: 4 secs:
35        //nullify of Imp wait
36        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(0));//sel 4.x
37        //e2: imp wait: 0
38        //e3: imp wait :0
39
```

And then again give value when needed.

```
39
40         //it wont work for non web elements: title, url, alert, windows
```

Explicitly wait-

Until is method.

Wait is interface.

Until is abstract.

Fluent wait class implements wait interface.

Webdriver wait extends fluent wait.

```
workspace1 - org.openqa.selenium.support.ui.WebDriverWai
WaitConcept.java   ImplicitlyWait.java   *ExplicitWait.java   WebDriverWait.class
19
20 import java.time.Clock;
21 import java.time.Duration;
22 import org.openqa.selenium.NotFoundException;
23 import org.openqa.selenium.TimeoutException;
24 import org.openqa.selenium.WebDriver;
25 import org.openqa.selenium.WebDriverException;
26 import org.openqa.selenium.WrapsDriver;
27 import org.openqa.selenium.remote.RemoteWebDriver;
28
29 /** A specialization of {@link FluentWait} that uses WebDriver instances. */
30 public class WebDriverWait extends FluentWait<WebDriver> {
31   private final WebDriver driver;
32
33   /**
34    * Wait will ignore instances of NotFoundException that are encountered (thrown) by de
35    * 'until' condition, and immediately propagate all others. You can add more to the ig
36    * calling ignoring(exceptions to add).
37    *
```

Eclipse File Edit Source Refactor Navigate Search Project Scout Run Window Help

workspace1 - org.openqa.selenium.support.ui.FluentWait

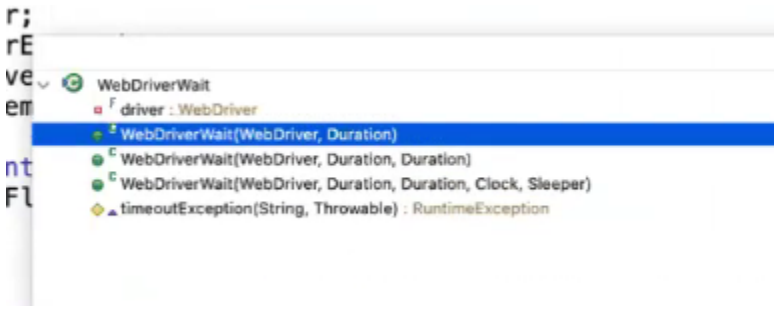
WaitConcept.java | ImplicitlyWait.java | *ExplicitWait.java | WebDriverWait.class | FluentWait.class

```
44  * <pre>
45  *   // Waiting 30 seconds for an element to be present on the page, checking
46  *   // for its presence once every 5 seconds.
47  *   Wait&lt;WebDriver&gt; wait = new FluentWait&lt;WebDriver&gt;(driver)
48  *       .withTimeout(Duration.ofSeconds(30L))
49  *       .pollingEvery(Duration.ofSeconds(5L))
50  *       .ignoring(NoSuchElementException.class);
51  *
52  *   WebElement foo = wait.until(new Function&lt;WebDriver, WebElement&gt;() {
53  *     public WebElement apply(WebDriver driver) {
54  *       return driver.findElement(By.id("foo"));
55  *     }
56  *   });
57  * </pre>
58  *
59  * <p><em>This class makes no thread safety guarantees.</em>
60  *
61  * @param <T> The input type for each condition used with this instance.
62  */
63 public class FluentWait<T> implements Wait<T> {
64
65   protected static final long DEFAULT_SLEEP_TIMEOUT = 500;
66
67   private static final Duration DEFAULT_WAIT_DURATION = Duration.ofMillis(DEFAULT_SLE
68
69   protected final T input;
70   protected final java.time.Clock clock;
71   protected final Sleeper sleeper;
72
```

WaitConcept.java | ImplicitlyWait.java | *ExplicitWait.java | WebDriverWait.class | FluentWait.class | Wait.class

```
1 // Licensed to the Software Freedom Conservancy (SFC) under one
17
18 package org.openqa.selenium.support.ui;
19
20 import java.util.function.Function;
21
22 /**
23  * A generic interface for waiting until a condition is true or not null. The condition may take a
24  * single argument of type .
25  *
26  * @param <F> the argument to pass to any function called
27  */
28 public interface Wait<F> {
29
30   /**
31    * Implementations should wait until the condition evaluates to a value that is neither null nor
32    * false. Because of this contract, the return type must not be Void.
33    *
34    * <p>If the condition does not become true within a certain time (as defined by the implementing
35    * class), this method will throw a non-specified {@link Throwable}. This is so that an
36    * implementor may throw whatever is idiomatic for a given test infrastructure (e.g. JUnit4 would
37    * throw {@link AssertionError}).
38    *
39    * @param <T> the return type of the method, which must not be Void
40    * @param isTrue the parameter to pass to the {@link ExpectedCondition}
41    * @return truthy value from the isTrue condition
42    */
43   <T> T until(Function<? super F, T> isTrue);
44 }
45
```

Control+o will give all methods of that class.
Webdriver wait has only one method in it-
Protected cant be accessed outside class. 34.00
No public methods inside webdriver wait.
Only three overloaded constructors available in webdriver wait.



workspace1 - org.openqa.selenium.support.ui.WebDriverWait - Eclipse IDE

WaitConcept.java | ImplicitlyWait.java | *ExplicitWait.java | WebDriverWait.class | FluentWait.class | Wait.class

```java
 * @see WebDriverWait#ignoring(java.lang.Class)
 */
public WebDriverWait(WebDriver driver, Duration timeout, Duration sleep) {
  this(driver, timeout, sleep, Clock.systemDefaultZone(), Sleeper.SYSTEM_SLEEPER);
}

/**
 * @param driver the WebDriver instance to pass to the expected conditions
 * @param clock used when measuring the timeout
 * @param sleeper used to make the current thread go to sleep
 * @param timeout the timeout when an expectation is called
 * @param sleep the timeout used whilst sleeping
 */
public WebDriverWait(
    WebDriver driver, Duration timeout, Duration sleep, Clock clock, Sleeper sleeper) {
  super(driver, clock, sleeper);
  withTimeout(timeout);
  pollingEvery(sleep);
  ignoring(NotFoundException.class);
  this.driver = driver;
}

@Override
protected RuntimeException timeoutException(String message, Throwable lastException) {
  WebDriver exceptionDriver = driver;
  TimeoutException ex = new TimeoutException(message, lastException);
  ex.addInfo(WebDriverException.DRIVER_INFO, exceptionDriver.getClass().getName());
  while (exceptionDriver instanceof WrapsDriver) {
    exceptionDriver = ((WrapsDriver) exceptionDriver).getWrappedDriver();
  }
  if (exceptionDriver instanceof RemoteWebDriver) {
    RemoteWebDriver remote = (RemoteWebDriver) exceptionDriver;
    if (remote.getSessionId() != null) {
      ex.addInfo(WebDriverException.SESSION_ID, remote.getSessionId().toString());
    }
    if (remote.getCapabilities() != null) {
      ex.addInfo("Capabilities", remote.getCapabilities().toString());
    }
  }
  throw ex;
}
}
```

Naveen website-

Login page.

Email field.

Password field.

Login button.

```java
package seleniumsessions;

import org.openqa.selenium.By;
import org.openqa.selenium.WebDriver;
import org.openqa.selenium.chrome.ChromeDriver;

public class ExplicitWait {

    public static void main(String[] args) {

        //ExplicitWait

        //Wait(I): until(); <---implements FluentWait(c): until(){}+other methods <---extends WebDriverWait(C): no methods

        WebDriver driver = new ChromeDriver();
        driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");

        By emailId = By.id("input-email");
        By password = By.id("input-password");
        By loginBtn = By.xpath("//input[@value='Login']");


    }
```

Webdriver wait has three constructors.

```
='Login']");
```

```
WebDriverWait(WebDriver driver, Duration timeout) - org.openqa.selenium.support.ui.WebDriverWait
WebDriverWait(WebDriver driver, Duration timeout, Duration sleep) - org.openqa.selenium.support.ui.WebDriverWait
WebDriverWait(WebDriver driver, Duration timeout, Duration sleep, Clock clock, Sleeper sleeper) - org.openqa.selenium.suppo
```

Expected conditions is class. Loads of static methods in it.

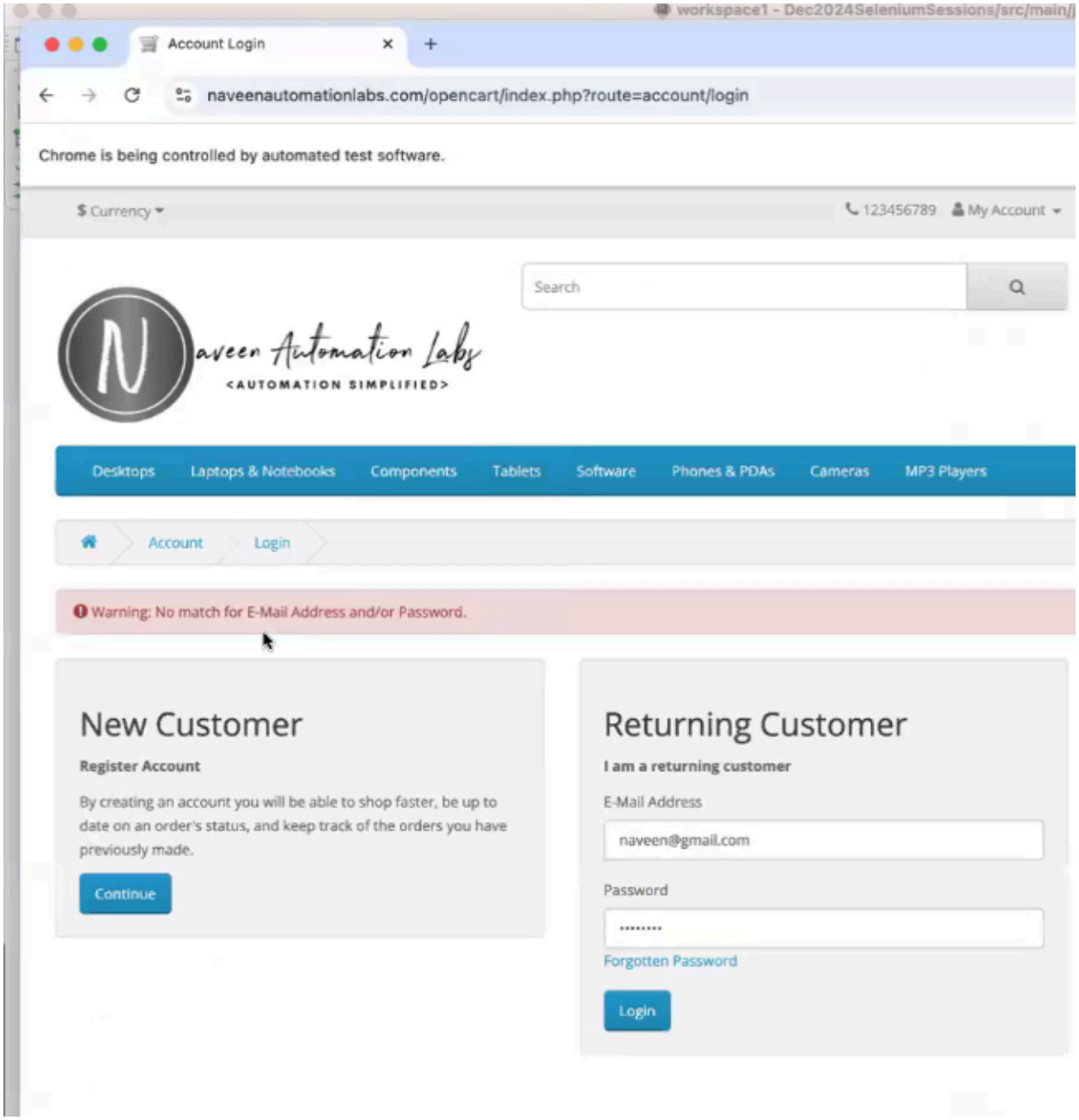
Until returns web element.

Wait for email-

```java
        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));//sel 4.x
        WebElement email_ele = wait.until(ExpectedConditions.presenceOfElementLocated(emailId));
        email_ele.sendKeys("naveen@gmail.com");
```

Password and login is normal as they must have been loaded by now(assumption)-

```java
        driver.findElement(password).sendKeys("test@123");
        driver.findElement(loginBtn).click();
```



What if we mix implicit and explicit-

```
19
20        WebDriver driver = new ChromeDriver();
21        driver.get("https://naveenautomationlabs.com/opencart/index.php?route=account/login");
22        driver.manage().timeouts().implicitlyWait(Duration.ofSeconds(20));//global wait = 20
23
24        By emailId = By.id("input-email");
25        By password = By.id("input-password");
26        By loginBtn = By.xpath("//input[@value='Login']");
27
28        WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));//sel 4.x
29        WebElement email_ele = wait.until(ExpectedConditions.presenceOfElementLocated(emailId));
30        email_ele.sendKeys("naveen@gmail.com");
31
32        driver.findElement(password).sendKeys("test@123");
33        driver.findElement(loginBtn).click();
34
```

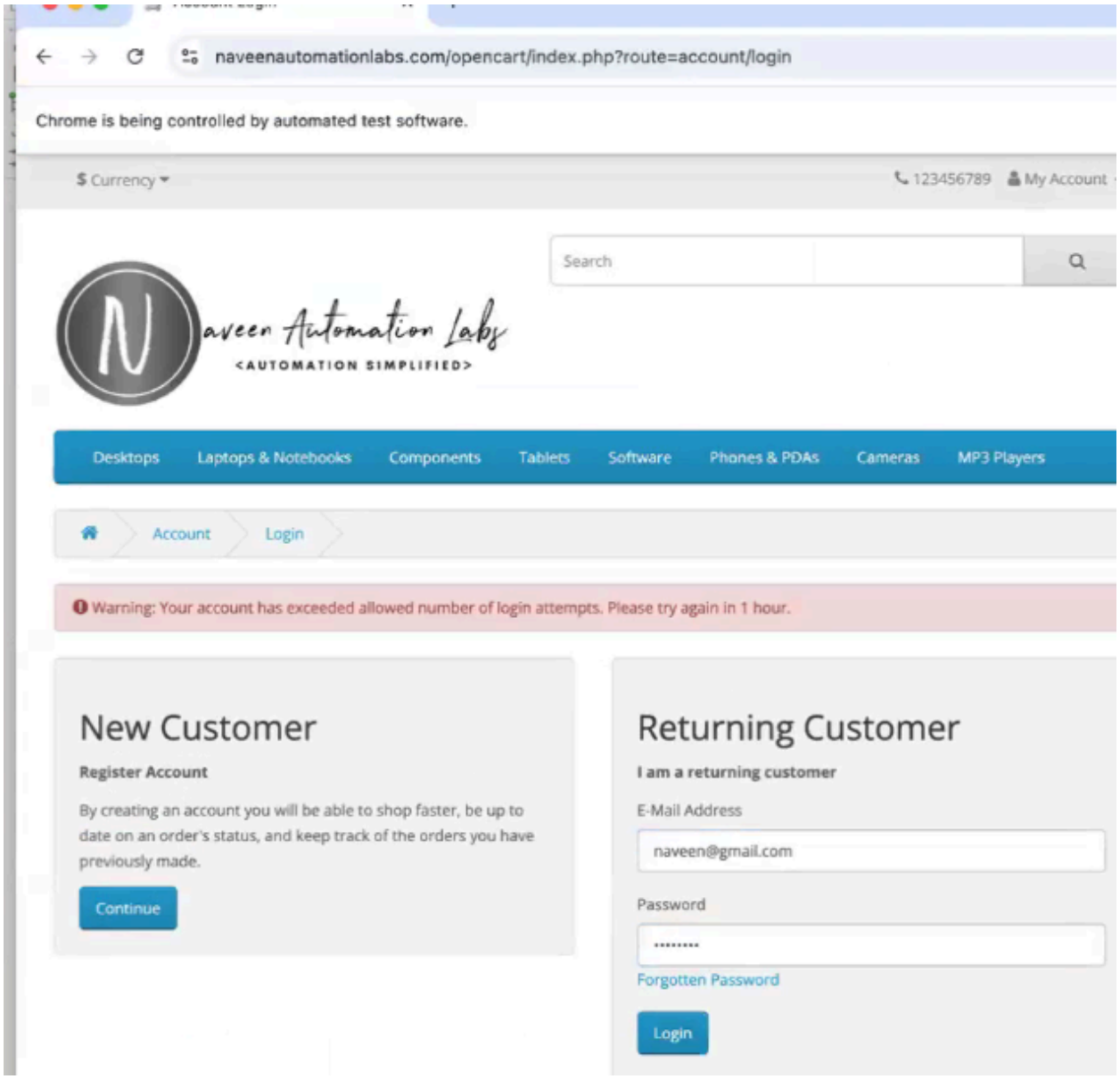
For password and login button it will wait for 20 seconds. For email first explicit wait will work then implicit wait will work for another 20 seconds.
Never mix implicity with explicit.
Create utility-
Added 14.
Updated 22.

```
10 import org.openqa.selenium.support.ui.WebDrive
11
12 public class ExplicitWait {
13
14     static WebDriver driver;
15
16     public static void main(String[] args) {
17
18         //ExplicitWait
19
20         //Wait(I): until(); <---implements Flu
21
22         driver = new ChromeDriver();
```

```
42
43     public static WebElement waitForElementPresence(By locator, int timeOut) {
44         WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));
45         return wait.until(ExpectedConditions.presenceOfElementLocated(locator));
46     }
47
48
49     public static WebElement getElement(By locator) {
50         return driver.findElement(locator);
51     }
52
```

Call the methods-
Same class.

```
36          waitForElementPresence(emailId, 10).sendKeys("naveen@gmail.com");
37          getElement(password).sendKeys("test@123");
38          getElement(loginBtn).click();
39
40      }
```



Give wrong email locator-

```
24
25          By emailId = By.id("input-email11");
26          By password = By.id("input-password");
27          By loginBtn = By.xpath("//input[@value='Login']");
28
```

After ten seconds gives this exception-

```
Console  Problems  Javadoc  Declaration  Progress  Results of running suite  Debug  Expressions  Terminal 1
<terminated> ExplicitWait (1) [Java Application] /Users/naveenautomationlabs/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20200815-0932/jre/bin/java (31-Mar-2025, 9:24:23 am – 9:24:39 am)
        at seleniumsessions.ExplicitWait.main(ExplicitWait.java:36)
Caused by: org.openqa.selenium.NoSuchElementException: no such element: Unable to locate element: {"method":"css selector","selecto
  (Session info: chrome=134.0.6998.166)
For documentation on this error, please visit: https://www.selenium.dev/documentation/webdriver/troubleshooting/errors#no-such-elem
Build info: version: '4.29.0', revision: '5fc1ec94cb'
System info: os.name: 'Mac OS X', os.arch: 'x86_64', os.version: '10.16', java.version: '14.0.2'
```

Give wrong email locator-

This time use normal get elements method.

Wont wait for ten seconds.

Immediate error.

```
24
25          By emailId = By.id("input-email11");
26          By password = By.id("input-password");
27          By loginBtn = By.xpath("//input[@value='Login']");
28
29 //       WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(10));//sel 4.x
30 //       WebElement email_ele = wait.until(ExpectedConditions.presenceOfElementLocated(emailId));
31 //       email_ele.sendKeys("naveen@gmail.com");
32 //
33 //       driver.findElement(password).sendKeys("test@123");
34 //       driver.findElement(loginBtn).click();
35
36          //waitForElementPresence(emailId, 10).sendKeys("naveen@gmail.com");
37          getElement(emailId).sendKeys("naveen@gmail.com");
38          getElement(password).sendKeys("test@123");
39          getElement(loginBtn).click();
40
41      }
```

```
Console  Problems  Javadoc  Declaration  Progress  Results of running suite  Debug  Expressions  Terminal 1
<terminated> ExplicitWait (1) [Java Application] /Users/naveenautomationlabs/.p2/pool/plugins/org.eclipse.justj.openjdk.hotspot.jre.full.macosx.x86_64_14.0.2.v20200815-0932/jre/bin/java (31-Mar-2025, 9:25:30 am – 9:25:35 am)
Mar 31, 2025 9:25:34 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find an exact match for CDP version 134, returning the closest version; found: 133; Please update to a Selenium
Exception in thread "main" org.openqa.selenium.NoSuchElementException: no such element: Unable to locate element: {"method":"css se
  (Session info: chrome=134.0.6998.166)
For documentation on this error, please visit: https://www.selenium.dev/documentation/webdriver/troubleshooting/errors#no-such-elem
Build info: version: '4.29.0', revision: '5fc1ec94cb'
System info: os.name: 'Mac OS X', os.arch: 'x86_64', os.version: '10.16', java.version: '14.0.2'
Driver info: org.openqa.selenium.chrome.ChromeDriver
Command: [6ab6537d2eb05f6cace60f0eca226a8f, findElement {using=id, value=input-email11}]
```

No auto wait in selenium.

Pw and cypress have auto wait.

Move to element util-

First dom comes then browser ui webelements.

```java
//Wait Utils**************//

/**
 * An expectation for checking that an element is present on the DOM of a page.
 * This does not necessarily mean that the element is visible.
 * @param locator
 * @param timeOut
 * @return
 */
public WebElement waitForElementPresence(By locator, int timeOut) {
    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));
    return wait.until(ExpectedConditions.presenceOfElementLocated(locator));
}
```

Dangerous method because if the element is not there on ui then error.

Best method-

Element util class.

```java
/**
 * An expectation for checking that an element is present on the DOM of a page and visible.
 * Visibility means that the element is not only displayed but also has a height and width that is greater than 0.
 * @param locator
 * @param timeOut
 * @return
 */
public WebElement waitForElementVisible(By locator, int timeOut) {
    WebDriverWait wait = new WebDriverWait(driver, Duration.ofSeconds(timeOut));
    return wait.until(ExpectedConditions.visibilityOfElementLocated(locator));
}
```

In any wait util you create, never give webelement as parameter. We are waiting for web element and if we pass as parameter that means there is no need to wait for it.

Click with wait-

Send keys with wait-

```java
public void clickWithWait(By locator, int timeOut) {
    waitForElementVisible(locator, timeOut).click();
}

public void sendKeysWithWait(By locator, int timeOut, CharSequence... value) {
    waitForElementVisible(locator, timeOut).sendKeys(value);
}
```

1.02 - char sequence to be the last method because its array and we need to pass atleast one value.

Overload get element with wait

```
134
135     public WebElement getElementWithWait(By locator, int timeOut) {
136         return waitForElementVisible(locator, timeOut);
137
138     }
```