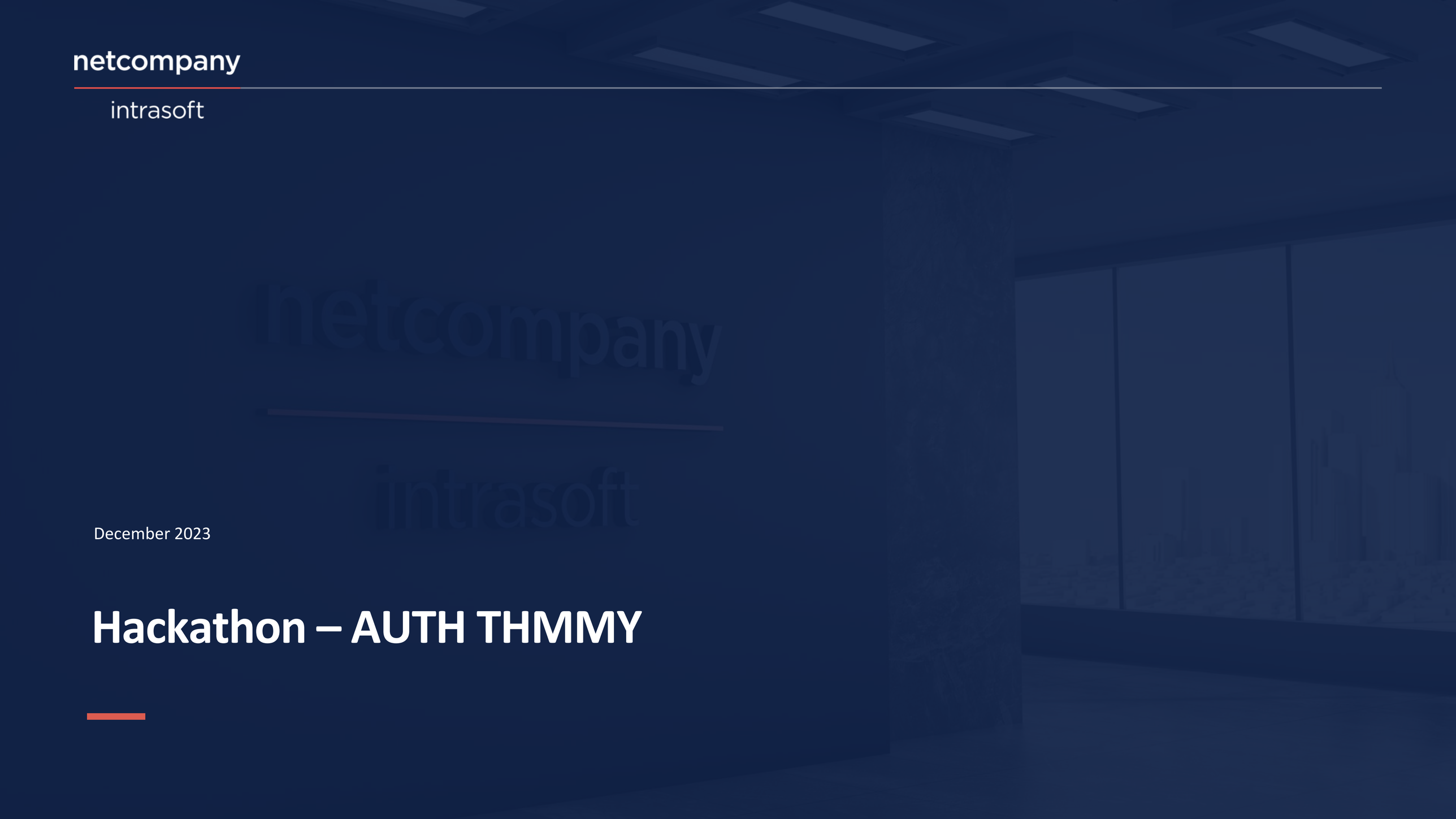
netcompany

intrasoft



December 2023

Hackathon – AUTH THMMY

netcompany
intrasoft

Topics

- Role of SW Testing
- Role of Test Analysis in STLC (SW Testing Lifecycle)
- Test Analysis Tasks
- Test Analysis & best practices
- Requirements
- Transformation of a Requirement to Test Case
- How to report a bug
- Hackathon Challenges
- UI Testing
- Classification of Browser Automation
- How Convert Test Case to Automation Script
- Introduction to Cypress
- Why Cypress?
- How to begin with Cypress
- Cypress Test Runner
- CI / CD & Benefits

netcompany

intrasoft

Test Analysis

Role of Testing

SW Testing main roles are:

- **Contributing to the quality of the software system:** When the defects found are corrected before the system is released for operational use.
- **Enhancing Software Reliability:** It assures that the software satisfies the highest requirements of performance and reliability by thoroughly analyzing test findings, giving end users confidence.
- **Pinpointing Weaknesses:** By identifying faults and vulnerabilities in software.
- **Informing Decision-Making:** Testing equips teams to make wise decisions regarding software enhancements, resource allocation, and deployment preparedness through data-driven insights.

Role of Test Analysis

Test Analysis is the process of gathering, analyzing and examining test artifacts or test data to create test scenarios or test cases by collecting specification requirements and develop test objectives.

In the dynamic life cycle of software development, Test Analysis plays a pivotal role by ensuring the:

- Quality
- Reliability
- Effectiveness

of the software being developed.

It is a crucial phase that occurs after the requirements gathering and before the software testing.

Finally, it optimizes Test Efficiency as it enables the development of focused and efficient test scenarios, maximizing the coverage of vital features while reducing needless testing.

Test Analysis and Design

The test analysis and design activity has the following major tasks:

- Review specifications requirements
- Evaluate testability
- Identify and prioritize
- Design test cases
- Identify test data needed for the execution
- Design test environment setup
- Traceability between requirements and test cases.

Best Practices for Test Analysis

netcompany
intrasoft

- Traceability Matrix
- Prioritize Test Cases
- Data-Driven Testing
- Regression Testing
- Performance Metrics
- Documentation and Reporting
- Automation Opportunities
- Adaptability & Flexibility

netcompany
intrasoft

Types and forms of Specification Requirements

Types:

- USs (User Stories)
- UCs (Use Cases)
- BPs (Business Processes Diagrams)
- BRs (Business Rules)
- State Diagrams
- XSDs (XML Schema Definition)
- …

Forms:

- Documents
- Jira Items
- Other tool Items
- …

netcompany intrasoft

Test Analysis Challenge

Application to Test:
GREENKART https://rahulshettyacademy.com/seleniumPractise/#/

In the first challenge you will practice the **Test Analysis** as you will have to create a Test Plan.

You will analyze **five** Specification Requirements, and you will have to prescribe detailed Test Cases that will cover the Test Strategy.

In addition, you will have to execute manually the Test Cases you have prescribed, ensuring the correct implementation, and locating any possible defects (bugs).

Closing this activity, you will have to perform some “free-style Testing” to locate any additional defects (bugs) existing in the application.

- Test Analysis (8pts) for the five User Stories + 1 pts for any new additional Test Case Prescribed
- Test Execution & “free-style Testing” (8pts)
- Defects (bugs) Reporting & Test Reporting (4pts)

netcompany
intrasoft

US1: Add Products in the Cart

User Story Name	Add Products in the Cart
User Story Description	As a user I want to add the products I want to buy in the cart
Actors	Users
Pre-Condition	System must be connected to the network
Acceptance Criteria(s)	• The products selected are included in the cart • The amount of money are calculated automatically and corrected • Multiple products can be selected

netcompany
intrasoft

US2: Complete Order for the products in the cart

User Story Name	Complete Order
User Story Description	As a user I want to create and complete the order for the products I have in the cart
Actors	Users
Pre-Condition	System must be connected to the network
Acceptance Criteria(s)	The order is correctly recorded, and a confirmation message is displayed

netcompany
intrasoft

US3: Search for Products

User Story Name	Search for Products
User Story Description	As a user I want to Search for the wanted products
Actors	Users
Pre-Condition	System must be connected to the network
Acceptance Criteria(s)	The products selected are the ones displayed in the search results' screen

netcompany
intrasoft

US4: Top Deals

User Story Name	Top Deals
User Story Description	As a user I want to see the top deals
Actors	Users
Pre-Condition	System must be connected to the network
Acceptance Criteria(s)	The list of the top deals is correctly displayed in the screen

netcompany
intrasoft

US5: Use Promo Codes/Coupons for an Order

User Story Name	Use Promo Code
User Story Description	As a user I want to use Promo Code for the order I have in the cart
Actors	Users
Pre-Condition	System must be connected to the network
Acceptance Criteria(s)	Inserting a promo code, the respective discount is automatically calculated

netcompany

intrasoft

Templates

Template

netcompany
intrasoft

Test Case

Test Case ID		
User Story Reference		
Test Case Title		
Test Case Description		
Prerequisite(s)		
Test Step ID	Action	Excepted Result

Template

netcompany intrasoft

Defect Report

Defect ID	
Defect Title	
Test Case ID	
Severity/Impact	
Defect Detailed Description	
Complementary Information	

Severity/Impact: Blocker – Critical – Major – Minor – Trivial

1. **Blocker**: Defective Functionality which prevents TT to proceed with any further execution on the application or most of its functionalities.
2. **Critical**: Defective Functionality which prevents TT to perform any execution for the area of the application.
3. **Major**: By Major, TT characterizes the defects located which are serious but do not prevent the rest of the execution in the area. The defect does not result in a failure, but causes the system to produce incorrect, incomplete, or inconsistent results, or the defect impairs the systems usability.
4. **Minor**: The Minor defect does not cause a failure, does not impair usability, and the desired processing results can be easily obtained by working around the defect.
5. **Trivial**: The Trivial defect is the result of non-conformance to a standard, is related to the aesthetics of the system, or is a request for an enhancement.

Template

netcompany
intrasoft

Test Report

Test Case ID	Date of Execution	Test Result*	Defect ID	Additional comments

Test Results:

- **Passed**: The actual result matches the expected result.
- **Failed**: The actual result does not match the expected result.
- **Blocked**: The test case cannot be executed (due to missing functionality, issue with the environment, ..)

netcompany

intrasoft

Test Automation

netcompany
intrasoft

Definition of UI Testing

UI testing is a software testing type that checks the Graphical User Interface of the Software to ensure that the functionalities of software application work as per specifications by checking screens and controls like menus, buttons, icons, etc.

netcompany
intrasoft

Classification of Browser Automation

Chrome DevTools Protocol (Non Standard)	Browser / Web Javascript API's (Non Standard)	WebDriver Protocol (Standard)
Playwright	Cypress	Selenium
Puppeteer	TestCafe	WebdriverIO

netcompany
intrasoft

How Convert Test Case to Automation Script

Test Case ID	check cypress tagging	
User Story Reference	describe()	
Test Case Title	it(‘’)	
Test Case Description	“Docstring comments in the class”	
Prerequisite(s)	before() or beforeEach() depending on the abstraction	
Test Step ID	Action	Excepted Result
	List/add etc	assert()

```
/// <reference types="Cypress" />
import ListContacts from '../pageObject/ListContacts';
import AddNewContact from '../pageObject/AddNewContact';
describe('Add A New Contact', function()
{
    before(function() {
        cy.request('DELETE', 'http://localhost:7001//api/v1/contacts/delete')

    })
    beforeEach(function() {
        cy.visit(Cypress.env('url') + "/#/contact");
        cy.fixture('example').then(function(data)
        {
            this.data = data
        })
    })
    it('Adding a new Contact with success', function() {
        const listContacts = new ListContacts();
        const addContact = new AddNewContact();
        //cy.visit(Cypress.env('url') + "/#/contact");
        listContacts.getNewContactUrl().click();
        addContact.getName().type(this.data.name);
        addContact.getEmail().type(this.data.email);
        addContact.getPhone().type(this.data.phone);
        addContact.getSaveBtn().click()
        listContacts.GetContactListTitle().should('have.text', 'Contact List')
        listContacts.VerifyEmailExists(this.data.email).should('be.visible');

    })

    it('Adding a Contact that already exists', function() {

        const listContacts = new ListContacts();
        const addContact = new AddNewContact();
        //cy.visit(Cypress.env('url') + "/#/contact")

        listContacts.getNewContactUrl().click();
        addContact.getName().type(this.data.name);
        addContact.getEmail().type(this.data.email);
        addContact.getPhone().type(this.data.phone);
```

netcompany
intrasoft

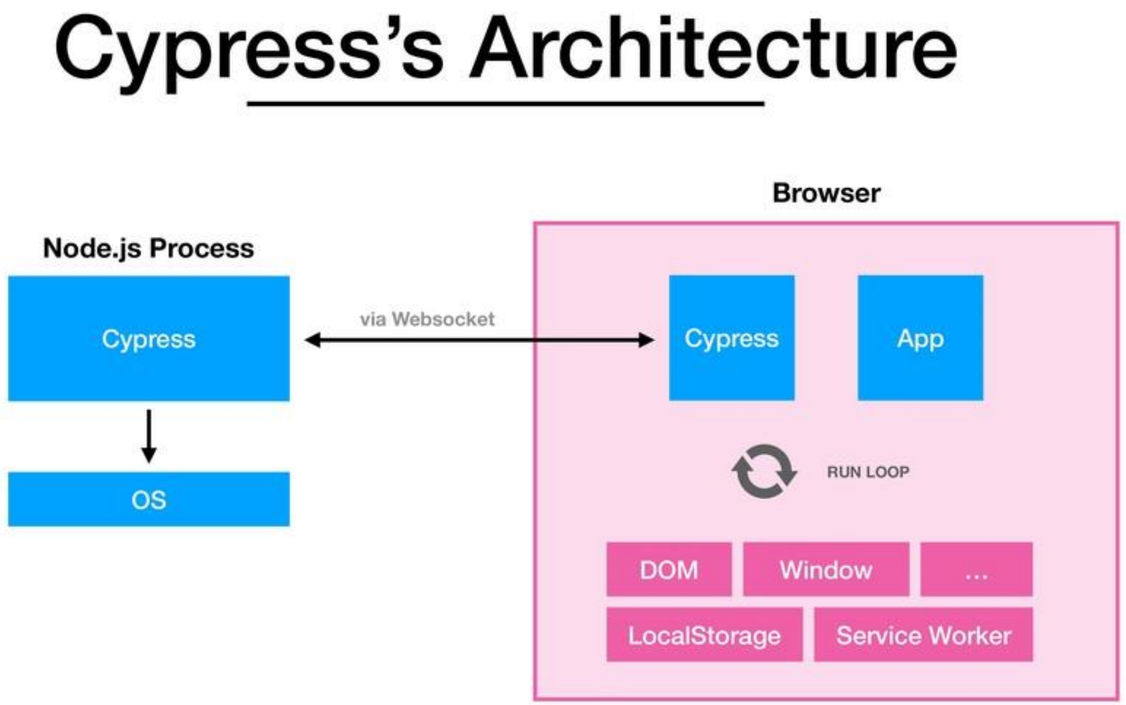
Introduction to Cypress

- Automation Testing Tool for the modern web applications.
- Fast, easy and reliable testing for anything runs in a browser.
- Basic understanding of JavaScript.

netcompany
intrasoft

Why Cypress?

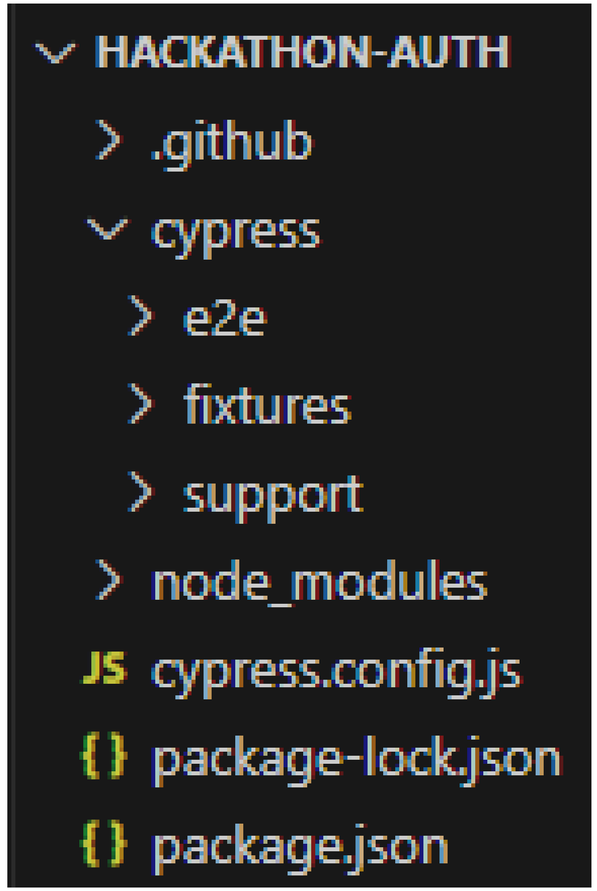
- Easy Installation
- Easy Development of Scripts
- Runs on real time
- Capability to “travel” in time
- Automatically waits for commands and assertions
- Debugging
- Stable and Reliable
- Active Community and Support


Cypress's Architecture
Browser
Node.js Process
Cypress
via Websocket
Cypress
App
OS
RUN LOOP
DOM
Window
...
LocalStorage
Service Worker

netcompany
intrasoft

A typical Cypress project

- fixtures: here we put our test data files (similar to resources)
- integration: all test scripts will be here only
- plugins: As cypress is also a node process, we can integrate multiple plugins here. (ie cucumber etc)
- support: Reusable methods or custom commands can be put here. (common-libs etc)



How to begin with Cypress

node.js and npm (node package manager)

Installation Steps:

1. npm –i init → create a package.json
2. npm install cypress - - save-dev → fetch latest cypress
3. node_modules/.bin/cypress open → open cypress runner

OR

1. npm install → if you clone a project

Example with Cypress for two main components

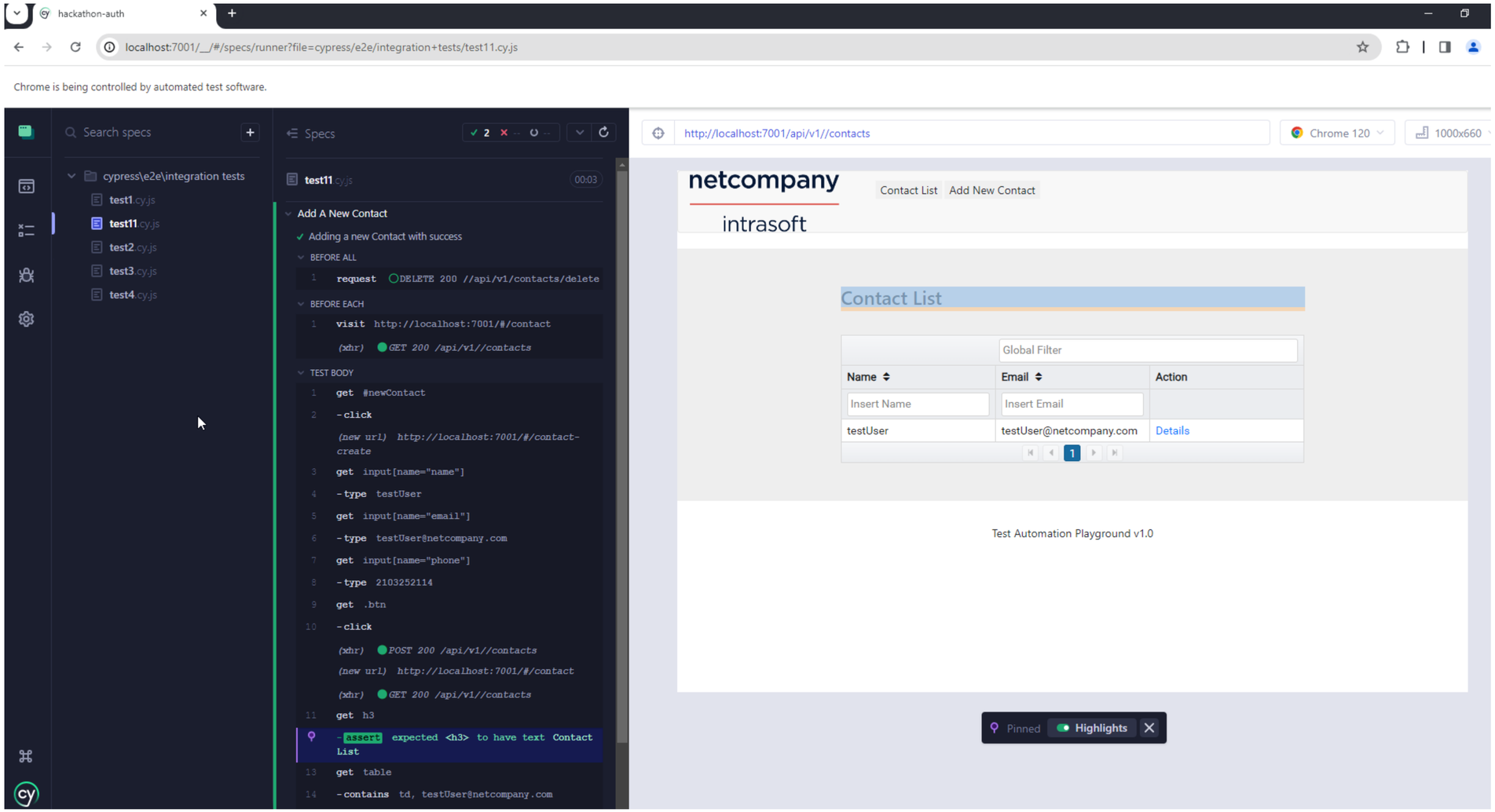
- Test Runner: Runs tests in a unique interactive runner that allows you to see commands as they execute while also viewing the application under test.

- Cypress Dashboard: Provides timely, simple and powerful insights on all your tests run at a glance.

Cypress Framework

netcompany
intrasoft

Cypress Test Runner



netcompany
intrasoft

Cypress assertions

- cy.get('li.selected').should('have.length', 3)
- cy.get('form').find('input').should('not.have.class', 'disabled’)
- cy.get('textarea').should('have.value', ‘hello world’)
- cy.get('[data-testid="user-name"]').should('have.text’, ‘Auth’)

https://docs.cypress.io/guides/references/assertions

```
/// <reference types="Cypress" />

import ListContacts from '../pageObject/ListContacts';
import AddNewContact from '../pageObject/AddNewContact';

describe('Add A New Contact', function()
{

    before(function() {
        cy.request('DELETE', 'http://localhost:7001//api/v1/contacts/delete')


    })

    beforeEach(function() {
        cy.visit(Cypress.env('url') + "/#/contact");

        cy.fixture('example').then(function(data)
        {
            this.data = data
        })
    })

    it('Adding a new Contact with success', function() {
        const listContacts = new ListContacts();
        const addContact = new AddNewContact();
        //cy.visit(Cypress.env('url') + "/#/contact");

        listContacts.getNewContactUrl().click();
        addContact.getName().type(this.data.name);
        addContact.getEmail().type(this.data.email);
        addContact.getPhone().type(this.data.phone);
        addContact.getSaveBtn().click()
        listContacts.GetContactListTitle().should('have.text', 'Contact List')
        listContacts.VerifyEmailExists(this.data.email).should('be.visible');

    })


    it('Adding a Contact that already exists', function() {

        const listContacts = new ListContacts();
        const addContact = new AddNewContact();
        //cy.visit(Cypress.env('url') + "/#/contact")

        listContacts.getNewContactUrl().click();
        addContact.getName().type(this.data.name);
        addContact.getEmail().type(this.data.email);
        addContact.getPhone().type(this.data.phone);
```

netcompany
intrasoft

CSS Locators

- id ---> #idname or tagname#idname
- classname ---> .classname or tagname.classname
- attribute ---> tagname[attribute='value’]
- tagnames ---> tagname1 tagname2

Useful Links

- https://docs.cypress.io/guides/overview/why-cypress
- https://cloud.cypress.io
- https://github.com/
- https://code.visualstudio.com/
- https://rahulshettyacademy.com/seleniumPractise/#/

netcompany
intrasoft

Test Automation Challenge

In the second challenge, you will practice Test Automation as you will have to develop an automation solution for prescribed Test Cases (from first challenge).

To be able to automate these Test Cases, you will have to use the https://www.cypress.io/ framework, you will prepare your code in JavaScript and you will put into practice some tips we presented before.

- Tools installation (2pts)
- TCs implementation and successful execution (up to 5pts per TC)

netcompany

intrasoft

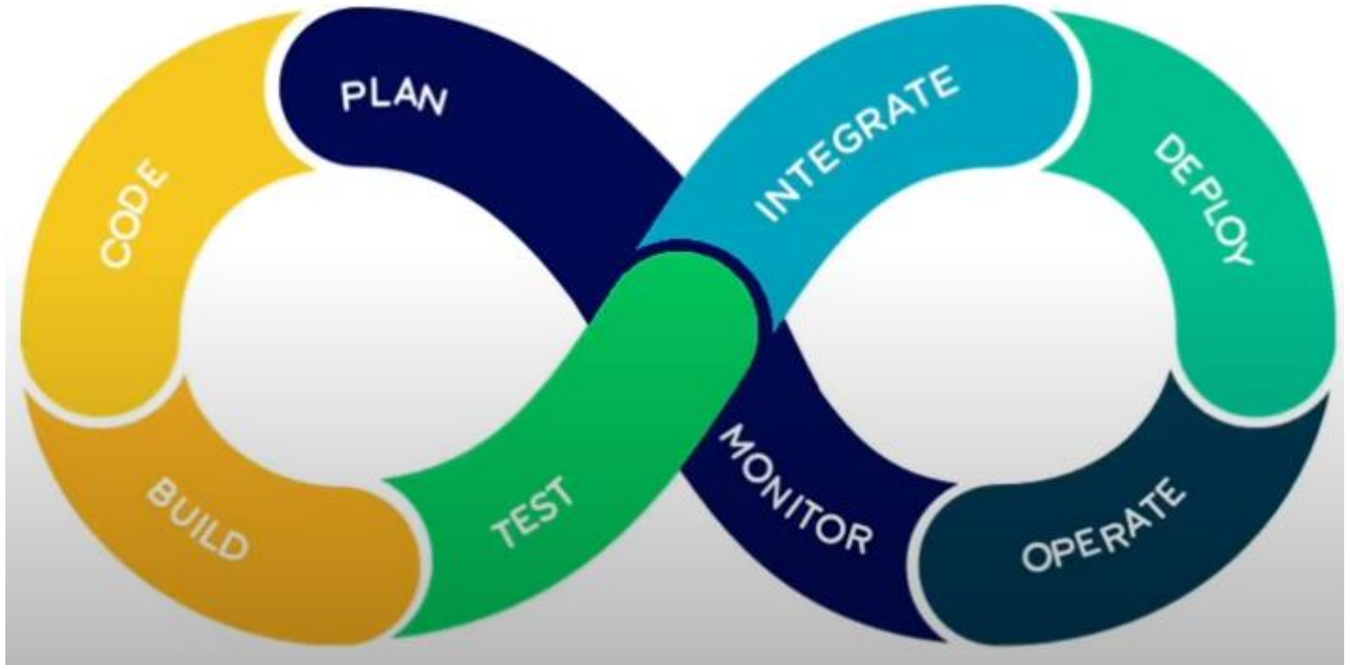
CI / CD

What is CI / CD

- Continuous integration (CI) is the practice, in software engineering, of merging all developer working copies to a shared mainline several times a day.

- Continuous delivery (CD) is a software engineering approach in which teams produce software in short cycles, ensuring that the software can be reliably released at any time. It aims at building, testing, and releasing software faster and more frequently.

DevOps phases



Benefits of CI / CD

- Reduced risk
- Increased confidence
- Better quality code
- Ready to ship code
- Systematic versioning
- Code quality trend analysis (check the development process, less or more bugs?)
- Time to market
- Reduced cost

TOOLS

Jenkins

Travis CI

TeamCity

Bamboo

GitLab

GitHub

netcompany
intrasoft

CI/CD Challenge

In the third challenge, you will practice the integration of the Automated Test solution within a CI/CD methodology as you will have to create your own GitHub pipeline.

The purpose of this pipeline will be to give the ability to someone to press a button and run the whole Testing Suite in an automated manner publishing eventually a Test Report (HTML) giving a detailed overview to the development team and testing team information about the successful/failed Tests, Test coverage etc. (Total: 10pts)

Practicalities

netcompany

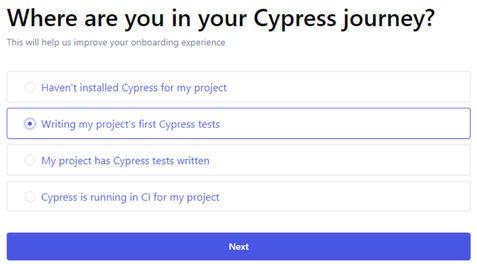
intrasoft

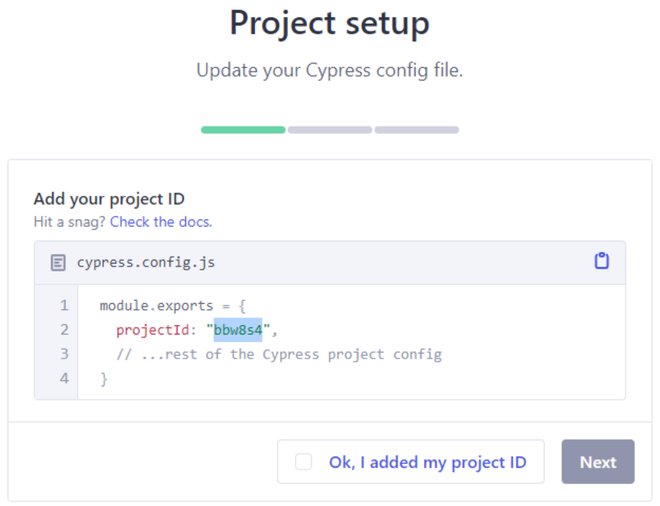
GitHub

- Teams structure: 3 to 5 members (currently **62** participants)
- Name your team however you love.
- **Each team (a chief team member) should create one Github project (name it however you love)** and invite team members to establish project collaboration among team members (git push/pull).
- For doing this https://docs.github.com/articles/inviting-collaborators-to-a-personal-repository
- Create locally a **cypress project** and push it to the main branch of your project
- Push also the test-analysis-template.xlsx under a folder named /test-analysis

cypress

- Integration with **https://cloud.cypress.io**
- Create a project in cloud.cypress.io (Organization name: better enter the name of your team). It works in similar fashion with Github (the chief team member setups the project and invites team members from the





- Fill in the email of other team members and click on “Send invites” button.
- After this step, you will be redirected in the project setup page where you can select to **add you project id** by enabling the corresponding checkbox. Then click on the “Next” button.
- MAKE sure to STORE the PROJECT ID

Tips for Challenges

Thank you!

Netcompany-intrasoft.com