**Zaynab Alsalih hkjajdlkKA**

1. **Eget val av produkt/tjänst att bygga upp automatisering mot. Skapa en dokumenterad testanalys av produkten/tjänsten (max 1 sida eller en mindmapp).**

<https://bibliotek.burlov.se/librarycontact>

Diagram

Description automatically generated

1. **Välja ett testverktyg för automatisering dokumentera hur valet av testverktyg gjordes (0,5-1 sida)**

Jag har kolllat på olika vertyger till exempel postman men det mest som jag tycker om att utveckla mig är selenium Ide och jag har frågat några testare som sa att det mest använder man postman och selenium i marknaden nu. Det är lättare att komma igång med och har bra verktyg vilket underlättar arbetet som testare. Som testare använder man testverktyg för API och GUI testning, till exempel SoapUI, Postman, Selenium och Cypress. Verktyget behöver bevisa att det verkligen kan lösa problemet och testa produkten på bästa sätt. Så jag Vald ett antal realistiska scenarion för att testa min produkt, och jag tycker att selenium Ide Klarar den på ett fördelaktigt och smidigt sätt.

Den nya Selenium IDE är utformad för att registrera interaktioner med webbplatser för att hjälpa att generera och underhålla webbplatsautomatisering, tester och ta bort behovet av att manuellt gå igenom repetitiva tagningar. Man kan:

\* Spela in och spela upp tester på Firefox och Chrome.

\* Organisera tester i sviter för enkel hantering.

\* Spara och ladda skript, för senare uppspelning.

\* Det behöver inte att man kunna ett programering språk.

\*Man behöver inte att bli en del av Teams för att kunna testa webbsidan med selenium Ide.

\*Jag behöver inte hitta dokumentation som postman behövde jag att hitta API dokumentation och den man kan inte hitta i alla webbsidor.

1. **Dokumentera hur automatiserade tester satts upp i testverktyget (t.ex hur du satte upp**

**testerna enligt ett specifikt format eller om du använder setup/teardown på ett visst sätt)**

**samt hur testdesignen ser ut för varje test (1-2 sidor).**

Det första test som jag gjorde vill jag testa att Webbsidan oppnas utan inge problemet sen testat jag click på serch för att jag vill se om click på serch fungerar och skickar mig till search fonister. Jag har använd target css=#navigation .search-link-icon för att clicka på search testen är ok.

Set Window size test: Testa om man kan oppna web windows på olika storlek till så forst oppnar jag webben och sen valde jag command set window size med olika target till exampel 1280x800 och sen testa om det kan bli mindre 300x200. Så jag vill kolla om den oppnas på olika storlek och testen var ok.

Close test : här vill jag testa om web windows stänger sig genom att använda commond close som behöver ingen target och testen var ok.

Select test : jag vill select olika spark alternativ och ser om den väljer rätt alternativ så jag vald command select. Jag cliker på språk först och sen väljer jag språk arabiska med sin label= arabiska sen byter jag och select svenska och test av select var ok.

Prova assert text command : jag har använd assert text för att jag vill prova och testa om texten skrivas rätt så jag provat om texten ”barn och ung” är rått sen jag vill assert text och ser om det öppetid för bibliotek från kl 7.00-11.00 och testen var ok.

När jag gjord assert text till adressen som skrev i web sidan. Anväd jag assert command och value för gatan var Lommavägen 2 men testen run visar det ett fel och value var Gata Lommavägen 2 mellan i websida av arlövsbibliotek skrev de bara Lommavägen 2.

Assert titel, jag har testat olika titel för att jag vill utvärdera uttryck och ett meddelande som ska skrivas ut om uttrycket är falskt. Om uttrycket är sant fortsätter programmet att köra som vanligt, men om det är falskt avbryts programmet och ett felmeddelande skrivs ut som indikerar vilket uttryck som inte var sant.

1. **Länk till ditt publika github repo ska finnas länkat till läraren inför rättning.**

[**https://github.com/zaynabalsalih?tab=repositories**](https://github.com/zaynabalsalih?tab=repositories)

1. **Dokumentation kring hur CI med hjälp av Jenkins sattes upp (max 1 sida)**

Continuous Integration (CI) med hjälp av Jenkins innebär att man göra en automatisk process för att bygga, testa en applikation varje gång det görs en förändring i källkoden. Detta hjälper till att säkerställa att koden är stabil och fungerar korrekt innan den skickas ut. För att sätta upp CI med Jenkins ska man först installera Jenkins.

Konfigurera Jenkins: öppna webbläsaren och gå till Jenkins-gränssnittet. Följ installationsguiden och skapa en ny användare.

Skapa en Jenkins-pipeline: Jenkins huvudmeny och välj "New Item" för att skapa en ny pipeline. Välj "Pipeline" som typ av projekt och ge det ett namn.

Ange pipeline-klon URL: I pipeline-inställningarna, ange Git-klon URL för applikation. Jenkins kommer att hämta koden från den angivna klonadressen.

Ange byggstegen: bygga, testa och distribuera applikationen. Exempel bygga koden, köra enhetstester, bygga ett distribuerbart paket, ladda upp det till ett moln eller skicka det till produktionsmiljön.

Spara pipeline-inställningarna: Spara pipeline-inställningarna och kör pipeline-processen. Jenkins kommer att utföra alla de konfigurerade stegen och skapa en rapport över resultaten.

Testa pipeline-processen: Testa pipeline-processen genom att göra en förändring i källkoden och pusha den till Git-klonen. Jenkins kommer automatiskt att starta pipeline-processen och rapportera resultatet.

Detta hjälper till att minska risken för buggar och problem i produktionen och säkerställer att applikationen fungerar som den ska.