Automatizovati testove za sajt <https://demoqa.com/>

Odvojite prvih 10-15 minuta u istrazivanju sajta, da proucite kako sta radi, onda pisite test kejseve u nekom dokumentu, zatim posle toga krenite da automatizujete te test kejseve. Napisite prvo 5 kejseva recimo pa ih automatizujte, zatim opet zapisete 5 pa ih automatizujete itd. Koristite POM, TestNG, Apache POI. Koristite sto vise komentara u kodu kao da na taj nacin objasnjavate nekome sta radite i zasto na taj nacin. Ako pisete neki test ostavite komentar zasto bas taj assert ili zasto ima vise assert-ova. Na pocetku u Base klasi napisite sta sve koristite za testiranje, koje alate, zasto bas te alate, u kom browseru, zasto bas u tom browseru, da li postoji drugi nacin da se nesto odradi, zasto ste odabrali bas ovaj itd. Napisite testove kako sajt TREBA da radi, a ne kako trenutno radi. Vi zadajte sebi zahteve, mogu biti kao trenutno stanje, nije nuzno da napravite nesto drugacije, ali ako vam se nesto ne dopada onda napisite test da radi kako vi smatrate da treba. Primera radi ako vam se ne dopada koja se notifikacija prikazuje, onda ubacite kako vi mislite da treba, test ce da padne i u kodu stavite komentar da ce taj test pasti iz tog i tog razloga.

Zapamtite da je ovo simulacija onoga sta cete raditi, ako naidjete na problem nemojte odmah traziti pomoc od drugih vec probajte sami da resite problem, mnogo vise cete nauciti na taj nacin. Koristite Google sto vise, koristicete neke stvari koje mozda nismo presli na predavanju (na primer odabir iz dropdown menija), istrazite sami sta mislite da vam treba. Cilj nije da automatizujete ceo sajt nego koliko stignete za ovu nedelju.

Jos jednom podsecam da pisete sto vise komentara u kodu jer ce se to mnogo ceniti kod vas kad vam budu gledali github nalog, ogroman plus dobijaju ljudi koji ostavljaju komentare i objasnjavaju code.

Ako vas buni koliko treba klasa za stranice da pravite to je na vama izbor, ali moj savet je recimo da gledate URL i kad god se promeni URL to je nova stranica.