

Kursa “Timekļa risinājumu automatizēta testēšana un slodzes testēšana” NightWatch.js moduļa noslēguma laboratorijas darbs

(Iesniegšanas termiņš 18/05 23:59)

Uzdevums: Izveido Nightwatch.js testa automatizācijas projektu, izmantojot Node.js. Projektam jābūt strukturētam pēc labākās prakses. Jāiestata paralēla testa izpilde, veicot testus pārlūkprogrammās Google Chrome un Firefox.

1) Uzdevums - automatizē lapu ar sekojošajiem scenārijiem:

<https://katalon-demo-cura.herokuapp.com/>

1. Scenārijs:

- a. Atver lapu
- b. Spied uz “Make Appointment”
- c. Ielogojies ar demo account
- d. Aizpildi sekojošos laukus:
- e. Facility: **Seoul** CURA Healthcare Center
- f. Apply for hospital readmission - **Checked**
- g. Healthcare Program: **Medicaid**
- h. Visit date: June 2 (02.06.2025)
- i. Comment: This is a test appointment
- j. Pārbaudi vai katra lauka dati ir pareizi, kad ir izveidots Appointment
- k. Izlogojies ārā
- l. Pārbaudi, ka izveidotā Appointment informācija vairs nav redzama

2. Scenārijs:

- a. Atver lapu
- b. Spied uz “Make Appointment”
- c. Ielogojies ar demo account
- d. Aizpildi sekojošos laukus:
- e. Facility: **Hongkong** CURA Healthcare Center
- f. Apply for hospital readmission - **NOT Checked**
- g. Healthcare Program: **Medicare**
- h. Visit date: January 1 (01.01.2026)
- i. Comment: Appointment in 2026
- j. Pārbaudi vai katra lauka dati ir pareizi, kad ir izveidots Appointment

- k. Izlogojies ārā
- l. Pārbaudi, ka izveidotā Appointment informācija vairs nav redzama

3. Scenārijs:

- a. Atver lapu
- b. Spied uz "Make Appointment"
- c. Ielogojies ar demo account
- d. Aizpildi sekojošos laukus:
- e. Facility: **Tokyo** CURA Healthcare Center
- f. Apply for hospital readmission - **Checked**
- g. Healthcare Program: **None**
- h. Visit date: July 15 (15.07.2025)
- i. Comment: Appointment in July
- j. Pārbaudi vai katra lauka dati ir pareizi, kad ir izveidots Appointment
- k. Izlogojies ārā
- l. Pārbaudi, ka izveidotā Appointment informācija vairs nav redzama

2) Uzdevums - izveido README.md failu, kurā ir:

- Īss projekta apraksts (tajā skaitā arī kādas Node.js un npm versijas tiek izmantotas un kā uzstādīt projektu)
- Nightwatch config faila un tajā esošo iestatījumu/settingu paskaidrojums
- Paskaidrojums kā var palaist:
 - 1 konkrētu testu
 - Visus testus
 - Testus konkrētā browserī
 - Testus paralēli

3) Uzdevums - izveido 1 testa failu, kurā:

- Saliec visus (vai cik uzrakstīji) testa scenārijus, ko rakstījām lekcijās vai tika rādīti prezentācijās patstāvīgam darbam (TDL School scenāriji, GitHub incorrect login scenārijs, Yahoo scenāriji)
- Katrs scenārijs ir atsevišķs tests
- Pielieto before/after beforeEach/afterEach hooks

NOSACĪJUMI UN INFO:

- Katram testam ir jāīsteno noteikta funkcionalitāte, un testi ir attiecīgi jāiedala testa failos
- 1. Uzdevumam jāizmanto POM (Page Object Model)
- Punkti tiks atskaitīti, ja tiek izmantots slikts kodēšanas stils — mainīgie ar “var”, nepareizas nosaukumu piešķiršanas metodes, nevienveidīgs funkciju deklarācijas stils (izņemot, ja varat norādīt, kāpēc tas bija jādara) utt.
- Nightwatch konfigurācijas failā jāiestata paralēla testu izpilde
- 3. Uzdevumam visi testi ir vienā failā un izmanto test hooks

UZDEVUMA IESNIEGŠANA UN VĒRTĒŠANA:

- Uzdevumu iesniegt formātā Vards_Uzvards_NightWatch.zip, saspiežot visus projekta failus un direktorijas vienā arhīvā, **izņemot node_modules direktoriju**.
- Maksimālais punktu skaits - 30 punkti, tajā ietilpst:
 - Page Object Model izmantošana (5 punkti)
 - Scenāriji ir automatizēti un izpildās pareizi, attiecīgi norādījumiem (5 punkti)
 - README fails ir izveidots ar paskaidrojumiem un atbildēm uz visiem jautājumiem (5 punkti)
 - Paralēla testa izpilde ir ieviesta un strādā (2 punkti)
 - Projekta struktūra (3 punkti)
 - Testi/kods ir strukturēts un viegli lasāms (3 punkti)
 - Lekcijās/patstāvīgi uzrakstītie scenāriji un to pareiza izpilde (5 punkti)
 - Test hooks izmantošana (2 punkti)