

**JAUNESNIOJO PROGRAMUOTOJO KVALIFIKACIJOS IV LYGIO**  
(įrašomas kvalifikacijos pavadinimas ir lygis pagal LTKS)  
**KOMPETENCIJŲ PRAKTINĖS DALIES (GEBĖJIMŲ) VERTINIMO UŽDUOTIS**

**UŽDUOTIS ORIENTUOTA Į VISĄ DARBO PROCESO CIKLĄ**  
**PRAKTINĖS DALIES METU VERTINAMOS KOMPETENCIJOS**

Kvalifikacijos vienetas (modulis)	Vertinamos kompetencijos
Informacinių sistemų projektavimas ir kūrimas (LTKS IV)	1.1. Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas.
Informacinių sistemų projektavimas ir kūrimas (LTKS IV)	1.2. Kurti tipinę programinę įrangą.
Nesudėtingų duomenų bazių projektavimas ir kūrimas (LTKS IV)	3.1. Projektuoti tipines reliacines ir nereliacines (NoSQL) duomenų bazes.
Programavimo aplinkos ir kūrimo proceso valdymas (LTKS IV)	4.3. Valdyti savo paties ir komandos atliekamą programinio kodo kūrimą.

**KOMPETENCIJŲ PRAKTINĖS DALIES (GEBĖJIMŲ) VERTINIMO UŽDUOTIS**

**Užduoties atlikimo trukmė: 5 val.**

Asmens įgytų kompetencijų praktinės dalies (gebėjimų) vertinimo užduoties aprašymas
<p>Sukurti WEB programą (angl. <i>application</i>), kurią pasitelkę miesto gyventojai ir svečiai galėtų rasti informaciją apie <b>mieste vykstančius renginius</b>, juos skelbti ir įvertinti.</p> <p><b>Programą (angl. <i>application</i>) sudaro dvi dalys:</b></p> <ul style="list-style-type: none"><li>• Administracinė sritis.</li><li>• Vieša sritis.</li></ul> <p><b>Administracinės srities aprašas ir jos privalomos funkcijos:</b></p> <ul style="list-style-type: none"><li>• Administracinė sritis turi būti sukurta naudojant NodeJS.</li><li>• Administracinės srities funkcijos prieinamos tik autentifikuotam vartotojui.</li><li>• Jeigu vartotojas nėra autentifikuotas, pradiniam puslapyje rodoma prisijungimo forma.</li><li>• Autentifikuotas vartotojas (Administratorius) gali patvirtinti renginį, blokuoti netinkamą turinį ir vartotojus, įvesti renginių kategorijas.</li></ul>

- Autentifikuotas vartotojas (Vartotojas) gali pridėti renginį nurodydamas pavadinimą, kategoriją, laiką, vietą, nuotrauką. Vartotojas gali koreguoti savo paskelbtus renginius, juos ištrinti.
- Administracinės srities vartotojo sąsajai gali būti naudojami Bootstrap šablonai.
- Visą informaciją iš administracinės srities į viešą sritį perduodama REST API.

#### **Viešos srities aprašas ir jos privalomos funkcijos:**

- Vieša sritis turi būti realizuota naudojant HTML5, CSS ir (arba) Bootstrap, JavaScript, React.
- Renginių paieška vykdoma naudojant filtrą pagal kategorijas ir renginio laiką.
- Renginiai gali būti atvaizduojami, kaip nurodyta viešos srities dizaino prototipe.
- Renginys įvertinamas, paspaudus žvaigždutę, jo kortelės dešinėje. Po paspaudimo įvertinimas išsaugomas duomenų bazėje.
- Renginio kortelės dešinėje, virš žvaigždutės, sumuojamas bendras reitingas (paspaudimų skaičius).

#### **Kompetencijų praktinės dalies (gebėjimų) vertinimo užduočiai atlikti reikalingi materialieji ištekliai**

- Operacinė sistema: Linux (rekomenduojama), MS Windows 10.
- Kodo redaktorius arba IDE (PHPStorm, VS code, Sublime text).
- Lokali programavimo aplinka: Apache arba Nginx WEB serveriu, MySQL duomenų bazių serveris (jei naudojama SQL), Git versijavimo sistema, Node JS, NPM.

<b>Užduoties etapai</b>
<b>Administracinės srities kūrimas:</b>
Sukurti naują projektą.
Sukurti duomenų bazę (jei naudojama SQL, suprojektuoti programos (angl. <i>application</i> ) modelius ir ryšius tarp jų).
Sukurti administratoriaus vartotojo sąsają.
Sukurti naujo renginio registracijos puslapį ir realizuoti renginio duomenų išsaugojimą duomenų bazėje. Formos duomenys turi būti validuojami, kad duomenų bazėje būtų korektiška informacija.
Sukurti renginių valdymo puslapį, kuriame būtų galima šalinti renginius ir keisti jų duomenis.
Sukurti API, kuris leidžia viešos srities vartotojui ieškoti informacijos apie renginius pagal dizaino prototipe nurodytus kriterijus ir juos įvertinti.
Sukurti <b>vartotojo registracijos</b> puslapį, kuriame vartotojas registruojasi, o jo duomenys išsaugomi duomenų bazėje. Pastaba: vartotojo slaptažodis negali būti saugomas duomenų bazėje atviru tekstu (plain text).
Sukurti <b>vartotojo prisijungimo</b> puslapį. Prisijungęs vartotojas autentifikuojamas ir autorizuojamas administracinėje sistemoje.

Suprojektuoti ir realizuoti viešos srities vartotojo sąsają. Privalomas suderinamumas su išmaniaisiais įrenginiais. Vartotojo sąsaja turi atitikti SPA (Single page application) koncepciją, t. y. visas funkcionalumas veikia viename puslapyje, o duomenys gaunami puslapio neperkraunant.
Realizuoti viešos srities funkcionalumą (renginio paieška ir įvertinimas).
Įkelti užduoties kodą į sukurta GITHUB saugyklą (angl. <i>repository</i> ). Turi būti sukurtas README.MD failas, kuriame surašyta instrukcija, kaip reikia paleisti programą (angl. <i>application</i> ), pateikti prisijungti prie administracinės ir kliento dalių reikalingi duomenys.

Vertinimo kriterijai	Įvertinimas (skiriamas balų skaičius)*
1. Duomenų bazės projektavimas.	2
2. Realizuotos užduotyje nurodytos administracinės dalies funkcijos.	3
3. Realizuotos užduotyje nurodytos kliento dalies funkcijos.	3
4. Programos (angl. <i>application</i> ) kodas patalpintas Github saugykloje (angl. <i>repository</i> ).	1
Refleksija	1

\*Nevisiškai pademonstravus gebėjimus rašoma pusė skiriamos balų sumos.  
 Nepademonstravus gebėjimų ar padarius esminę klaidą balai neskiriami.  
 Praktinės dalies (gebėjimų) vertinimo užduotis vertinama pagal dešimties balų sistemą.  
 Refleksijai (įsivertinimui) skiriama 10 proc. bendro užduoties įverčio.