**JAUNESNIOJO PROGRAMUOTOJO (JAVA PROGRAMUOTOJO)**

**KVALIFIKACIJOS IV LYGIO**

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

**UŽDUOTIS ORIENTUOTA Į VISĄ DARBO PROCESO CIKLĄ**

**PRAKTINĖS DALIES METU VERTINAMOS KOMPETENCIJOS**

|  |  |
| --- | --- |
| **Kvalifikacijos vienetas / modulis** | **Vertinamos kompetencijos** |
| 1. Informacinių sistemų projektavimas ir kūrimas (LTKS IV) | 1.1 Projektuoti ir programuoti žiniatinklio puslapių vartotojo sąsajas. |
| 1. Informacinių sistemų projektavimas ir kūrimas (LTKS IV) | 1.2. Taikyti programinės įrangos kūrimui naudojamus informatikos principus ir metodus. |
| 1. Informacinių sistemų projektavimas ir kūrimas (LTKS IV) | 1.4. Kurti tipinę programinę įrangą. |
| 2. Nesudėtingų duomenų bazių projektavimas ir kūrimas (LTKS IV) | 2.1. Projektuoti tipines reliacines ir nereliacines (NoSQL) duomenų bazes. |
| 3. Programavimo aplinkos ir kūrimo proceso valdymas (LTKS IV) | 3.3. Valdyti savo paties ir komandos atliekamą programinio kodo kūrimą. |

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

|  |
| --- |
| **Asmens įgytų kompetencijų praktinės dalies (gebėjimų) vertinimo užduoties aprašymas** |
| **Užduoties atlikimui skirtas laikas: 5 val.**  Sukurti aplikaciją, kurios pagalba automobilių servisų klientai galėtų rasti informaciją apie automobilių remonto servisų meistrus bei juos įvertinti.  **WEB aplikaciją sudaro dvi dalys:**   * Administracinė * Kliento   **Administracinė dalies aprašymas ir jos privalomos funkcijos:**   * Administracinė dalis turi būti sukurta naudojant JAVA programavimo kalbą, taikant objektinio programavimo principus (gali būti naudojamas JAVA programavimo kalbos karkasas). * Administracinės dalies projektavimas privalo remtis MVC architektūra. * Administracinės dalies funkcijos prieinamos tik autentifikuotam vartotojui, kuris autorizuojamas sistemoje pagal rolę. * Jeigu vartotojas nėra autentifikuotas, sistemos pradžioje rodoma prisijungimo forma. * Autentifikuotas vartotojas gali:   + Pridėti (nurodydamas serviso pavadinimą, adresą, vadovą), redaguoti, pašalinti servisą.   + Pridėti (nurodydamas: vardą, pavardę, specializaciją, miestą bei pasirinkdamas servisą, kuriame meistras dirba), redaguoti, pašalinti meistrą. * Duomenys turi būti saugomi suprojektuotoje ir sukurtoje duomenų bazėje (MySQL, MariaDB, H2 ir kt.). * Vartotojo sąsaja gali būti realizuota naudojant CSS, JavaScript karkasus ir bibliotekas. * Vartotojo sąsaja turi būti estetiška ir funkcionali.   **Kliento dalies aprašymas ir jos privalomos funkcijos:**   * Klientas gali įvertinti meistrą. Meistro įvertinamas išsaugomas duomenų bazėje. * Prie meistro rodomas bendras reitingas (visų įvertinimų vidurkis).   **Užduoties pateikimas:**   * Užduoties kodas turi būti patalpintas www.github.com sukurtoje repozitorijoje. * Moksleivis užbaigęs užduotį turi vertintojui nusiųsti github repozitorijos adresą. |
| **Kompetencijų praktinės dalies (gebėjimų) vertinimo užduočiai atlikti reikalingi materialieji ištekliai** |
| * Operacinė sistema: Linux (rekomenduojama), MS Windows 10 * Kodo redaktorius arba IDE (IntelliJ, Eclipse, NetBeans, Sublime text) * Lokali programavimo aplinka su: JAVA interpretatoriumi, Apache arba Tomcat serveriu, MySQL duomenų bazių serveriu, Git versijavimo sistema. |

|  |
| --- |
| **Užduoties etapai** |
| Sukurti projekto struktūrą. |
| Suprojektuoti aplikacijos modelius ir ryšius tarp jų. |
| Suprojektuoti ir sukurti duomenų bazę. |
| Sukurti užduotyje nurodytus funkcionalumus. |
| Sukurti vartotojo registraciją. Vartotojo duomenys išsaugomi duomenų bazėje. Pastaba: vartotojo slaptažodis, negali būti saugomas duomenų bazėje atviru tekstu (plain text). |
| Sukurti vartotojo prisijungimą, kurio pagalba vartotojas autentifikuojamas administracinėje sistemoje. |
| Įkelti užduoties kodą į sukurtą GITHUB repozitoriją. Turi būti sukurtas README.MD failas, kuriame surašyta instrukcija kaip reikia pasileisti aplikaciją, pateikti prisijungimui prie administracinės ir kliento dalių reikalingi duomenys. |

|  |  |
| --- | --- |
| **Vertinimo kriterijai** | **Įvertinimas (skiriamas balų skaičius)\*** |
| 1. Administracinė aplikacijos dalis atitinka MVC architektūra ir remiasi objektinio programavimo principais\*\*.  2. Duomenų bazės projektavimas  3. Realizuotos užduotyje nurodytos administracinės dalies funkcijos.  4. Realizuotos užduotyje nurodytos kliento dalies funkcijos.  5. Aplikacijos kodas patalpintas github. | 1  1  3  3  1 |
| Refleksija | 1 |

\* Nepilnai pademonstruotas vertinimo kriterijus – rašoma pusė skiriamos balų sumos.

\*\* Neįgyvendinus kriterijaus balas neskiriamas.

Praktinės dalies (gebėjimų) vertinimo užduotis vertinama pagal dešimties balų sistemą. Refleksijai / įsivertinimui skiriama 10 proc. bendro užduoties įverčio.