|  |
| --- |
| II variantas  **KOMPETENCIJŲ VERTINIMO UŽDUOTIS:**  **„TINKLARAŠTIS“**  **Užduoties projekto paruošimas**  Egzaminuojamajam pateikiamas *maven* projekto šablonas. Šį šabloną galima naudoti kaip pagrindą užduoties scenarijuje aprašytų reikalavimų realizavimui. Šablone esančio projekto sandara bei paleidimo informacija pateikta priede Nr. 1.  Atsiskaitymui (gynimui) pateikiamas darbas turi būti veikianti Java programėlė, supakuota kaip *.jar* arba *.war* failas, bei programėlės *source* kodas.  Kodo projektas turi būti suarchyvuotas (zip formatu) kartu su visais reikalingais resursais (duombaze, nustatymų failu) ir patalpintas viešai Internetu prieinamoje vietoje (rekomenduojama naudoti Google Drive). Nuoroda į archyvą turi būti įrašyta į egzamino vertinimo komisijos pateiktą dokumentą.  **Nefunkciniai reikalavimai:**  Atsiskaitomasis projektas:   * vykdomas kaip .*jar* arba .*war* failas ***DONE*** * surenkamas *maven* įrankiu iš *source* kodo ***DONE*** * realizuotas kaip internetinė programėlė ***DONE*** * naudoja duomenų bazę ***DONE*** * realizuoja visas *CRUD* operacijas *HTTP* ir duomenų bazės lygmenyse ***DONE***   Papildomų technologijų naudojimas nėra draudžiamas.  Reikalavimai kodo kokybei:   * programinės įrangos kodas turi būti tvarkingai atitraukiamas, kintamieji vadinami pagal Java įprastas taisykles bei kiek įmanoma laikomasi kitų Java programavimo standarto taisyklių; ***DONE*** * klasės turi būti logiškos ir atlikti tik joms numatytas funkcijas ***DONE*** * kodas, kur reikia, turi būti dokumentuojamas JavaDoc formatu; ***DONE*** * kodas turi būti padengtas modulių (*unit*) testais; * programos įvykiai turi būti kaupiami įvykių žurnale *(angl. log).* ***DONE***   **Funkciniai reikalavimai:**  **KOMPETENCIJŲ VERTINIMO UŽDUOTIS:**  **„TINKLARAŠTIS“**  Sukurkite programinę įrangą skirtą tinklarasčio (blogo) rašymui.  Programinės įrangos paskirtis - leisti tinklarasčio administratoriui rašyti naujus įrašus. Vartotojams galima juos skaityti, įvertinti bei palikti komentarus/klausimus, į kuriuos administratorius gali pateikti atsakymus.  Pagrindiniai funkciniai reikalavimai programinei įrangai:   * turi vaizduoti visus tinklarasčio įrašūs ***DONE DONE*** * turi leisti vartotojui peržiūrėti individualius įrašus ***DONE DONE*** * turi leisti vartotojui palikti komentarus po įrašų ***DONE DONE*** * turėti administratoriaus puslapį: ***DONE***   + leidžiantį administratoriui pridėti naują įrašą į tinklaraštį ***DONE DONE***   + leidžiantį administratoriui ištrinti esamą įrašą (kartu su komentarais) ***DONE***   + leidžiantį administratoriui redaguoti esamus įrašus ***DONE***   + leidžiantį administratoriui atsakyti į komentarą (atsakymas bus rodomas šalia komentaro, vienas komentaras gali turėti vieną atsakymą) ***DONE***   *Praktiškai administratoriaus puslapis turėtų būti apsaugotas ir prieinamas tik žinant vartotojo duomenis. Tačiau, kadangi internetinių puslapių saugumo tema nebuvo dėstoma, puslapis yra atviras visiems.* ***Bonus taškai****:už administratoriaus puslapio apsaugojimą - jei jis yra prieinamas tik žinant slaptažodį.*  Bendrieji reikalavimai:   * puslapio stilius nebus vertinamas, tačiau jis turi būti aiškus ir patogus naudoti * puslapio tekstas yra vientisas - parašytas viena kalba, be gramatinių klaidų * įvykusių klaidų žinutės yra suprantamos vartotojui |
|
|
|
|
|
|
|
| **Priedas Nr. 1**  **Užduoties projekto šablonas**  Kartu su užduotimi pateikiamas užduoties projekto šablonas.  Užduoties projektas - *maven* sąrankos įrankiui skirtas projektas. Projekte integruotos visos užduotyje reikalaujamos naudoti technologijos. Projekte taip pat yra programinės logikos pavyzdžių bei vartotojo sąsajos šablonų. Visais artefaktais, esančiais projekte, galima remtis atliekant scenarijuje pateiktą užduotį.  **Sąranka**  Projektą galima surinkti naudojant komandą:  *mvn clean package*  Komanda paruoš veikiantį *jar* failą, kurį vykdyti galima su komanda *java -jar*  Projektą kaip web aplikaciją paleisti galima naudojant komandą:  *mvn spring-boot:run*  Pagal nutylėjimą **React** *front-end* dalis bus pasiekiama per *localhost:8080* adresą. Todėl **rest** resursai turėtų būti pasiekiami kitu adresu, pvz. *localhost:8080/api*.  Projekto kūrimo metu rekomenduojama startuoti *front-end* ir *back-end* dalis atskirai. Pvz. vykdant *npm start* komandą iš *src/main/app* direktorijos, norint paleisti front-end dalį, o back-end dalį paleisti per *IntelliJ* kaip *spring boot* projektą  **Duomenų bazė**  Projektas naudoja integruotą H2 duomenų bazę. Pagal nutylėjimą projekte sukonfigūruota duomenų bazė saugoma atmintyje. Failų *src/main/resources/data.sql* ir *src/main/resources/schema.sql* pagalba, kievkieną kartą kai yra startuojama programėlė, duomenų bazėje sukuriamos bei užpildomos lentelės. Pavyzdys, kaip prisijungti prie duomenų bazės ir kaip joje vykdyti užklausas, yra sisteminiame kode.  **Struktūra**  Projekto konfigūracijos failas patalpintas *src/main/resources/application.properties*. Front-end dalis patalpinta *src/main/app*, back-end dalis patalpinta *src/main/java*. Pavyzdiniai komponentai *GreetingController*, *GreetingsDao*.  *Projekto struktūra, bei naudojamos technologijos tėra pasiūlymai, galima naudoti visiškai kitas technologijas, kol yra įvykdomi projekto reikalavimai.* |
|
|
|
|