RTU Liepājas akadēmija

Dabas un inženierzinātņu centrs

Kārlis Lācītis,

Profesionālā bakalaura studiju programmas   
„Informācijas tehnoloģija”   
2.kursa students

**“Code Moodle”**

Studiju darbs

Zinātniskais vadītājs –   
Lekt. Dzintars Tomsons

Liepāja 2025

**Anotācija**

**Darba autors:** Kārlis Lācītis

Darba tēma: “Code Moodle”

**Darba veids:** Studiju darbs

**Studiju programma:** Informācijas tehnoloģija

**Darba zinātniskais vadītājs:** Lekt. Dzintars Tompsons

**Darba apjoms:** 23 lpp.

**Atslēgas vārdi:** datubāzu pārvaldības sistēmas, programmatūras inženierija, datorsistēmu admninistrēšana, programmatūras projekta pārvaldība

**Pētījuma mērķis:** Noteikt piemērotāko datubāžu pārvaldības sistēmu veidojamam projektam.

**Darba saturs:** Darbs sastāv no 2 daļām.

Darba I daļa ir autora teorētiskais pētījums par piemērotāko datubāžu pārvaldības sistēmu veidojamā studiju projekta darbam. Daļa sastāv no ievada, datubāžu vēsturi, vērtēšanas kritēriju aprakstu, datubāžu aprakstu un vērtējumus, secinājumus un izmantotās literatūras saraksta. 1.nodaļa satur aprakstu par datubāzu uzbūvi un datubāzu pārvaldības sistēmu attīstības vēsturi. 2.nodaļā iekļauts iekļauts autora izvēlēto datubāžu pārvaldības sistēmu apskats un to salīdzinošās analīze – novērtēšanas kritēriju pamatojums, programmatūras novērtējums pēc izvēlētajiem kritērijiem un analīzes rezultātu apraksts.

Darba II daļa ir komandas projekta „Code Moodle” izstrādātās programmatūras apraksts. 1.nodaļā tiek raksturots programmatūras izstrādes projekts. 2.nodaļā dotas projekta nodevumu apraksts. 3.nodaļa raksturo komandas dalībnieku un autora ieguldījumu komandas projektaīstenošanā.

**Pētījuma metodes:** Literatūras studijas, datubāzu pārvaldības sistēmu salīdzinošā analīze, izvēlētās datubāžu pārvaldības sistēmu aprobācija.

**Darba rezultāti:** Datubāzu pārvaldības sistēmu salīdzinošā analīze, programmatūras projekta apraksts, programmatūras projekta dokumentācijas apraksts.

**Darba izmantojamība:** Projektā izstrādāto platformu paredzēts izmantot kodēšanas darbu piešķiršans, veikšanu un labošanas atvieglošanai skolotājiem un to studentiem. Darba I daļu var izmantot studenti un citi interesenti pamatzināšanau ieguvei par datubāzu uzbūvi un kādiem projektiem tie ir piemēroti. Savukārt raksturotos datubāzu pārvaldības sistēmu novērtēšanas kritēriju var izmantot līdzīgu projektu izstrādē programmatūras rīku izvēles noteikšanai.

**Annotation**

**Author of the paper:** Kārlis Lācītis

**Theme of the paper:** Databases, software development

**Type of the paper:** Annual project

**Study program:** Information Technology

**Consultant of the paper:** Lecturer Dzintars Tomsons

**Volume of the paper:** 23 pages

**Key words:** Database management systems, software engineering, computer system administration, software project management

**Aim of the research**: Determine the most suitable database management system for the project being developed.

**Content of the paper:** The work consists of two parts.

Part I is the author's theoretical research on the most suitable database management system for the proposed study project. This part includes an introduction, a history of databases, a description of evaluation criteria, database descriptions and evaluations, conclusions, and a list of references. Chapter 1 contains a description of database structure and the history of database management system development. Chapter 2 includes an overview of the author's chosen database management systems and their comparative analysis – justification of evaluation criteria, software evaluation based on the selected criteria, and a description of the analysis results.

Part II describes the software developed for the "Code Moodle" team project. Chapter 1 characterizes the software development project. Chapter 2 provides a description of the project deliverables. Chapter 3 describes the contributions of the team members and the author to the implementation of the team project.

**Methods of research:** Literature review, comparative analysis of database management systems, approbation of the selected database management system.

**Results of the research:** Comparative analysis of database management systems, software project description, software project documentation description.

**Applicability of the paper:** The platform developed in this project is intended to simplify the assignment, execution, and correction of coding tasks for teachers and their students. Part I of the work can be used by students and other interested individuals to gain fundamental knowledge about database structures and the types of projects they are suitable for. Furthermore, the described database management system evaluation criteria can be applied in similar projects to determine the selection of software tools.

Saturs

[I daļa Individuālais pētījums, piemērotākā datubāze projektam “Code Moodle” 5](#_Toc199030279)

[Ievads 5](#_Toc199030280)

[1. Datubāžu pārvaldības sistēmas un to attīstības vēsture 5](#_Toc199030281)

[1.1. Kas ir datubāžu pārvaldības sistēma 5](#_Toc199030282)

[1.2. Datu bāžu vēsture 5](#_Toc199030283)

[2. Datubāzu pārvaldības sistēmu salīdzinošā analīze 5](#_Toc199030284)

[2.1. Vērtēšanas kritēriji 5](#_Toc199030285)

[2.2. Datubāzu pārvaldības sistēmas 9](#_Toc199030286)

[2.3. Datubāzu pārvaldības sistēmu salīdzinošā analīze 14](#_Toc199030291)

[2.4. Analīzes rezultāti 14](#_Toc199030292)

[Secinājumi 14](#_Toc199030293)

[Jēdzienu skaidrojumi un saīsinājumi 15](#_Toc199030294)

[Izmantotā literatūra 16](#_Toc199030295)

[II daļa <(programmatūras projekts, komandas veikums)> 17](#_Toc199030296)

[1. “Code Moodle” projekta apraksts 17](#_Toc199030297)

[1.1 Projekta mērķis un uzdevumi 17](#_Toc199030298)

[1.2 Projekta organizatoriskā struktūra 17](#_Toc199030299)

[1.3 Projekta nodevumi 17](#_Toc199030300)

[2. Projekta nodevumu apraksts 18](#_Toc199030301)

[3. Autora un pārējo komandas dalībnieku ieguldījums projekta tapšanā 18](#_Toc199030302)

# I daļa Individuālais pētījums, piemērotākā datubāze projektam “Code Moodle”

## Ievads

Projekta ietvaros ir nepieciešams izmantot kādu no datubāžu pārvaldības sistēmām (DBVS) lai glabātu lietotāju datus un lietotāju ievaddatus. DBVS klāsts ir plašs un ir nepieciešams izprast pieejamo DBVS iespējas un to piemērotību veidojamai sistēmai.

Šī darba aktualitāte ir nepieciešamība izvēlēties vispiemērotāko DBVS veidojamam projektam, ņemot vērā to veiktspēju, funkcionalitāti un mērogojamību. Tā kā ir pieejamas gan relāciju, gan nerelāciju (NoSQL) datubāzes, katra ar dažādām priekšrocībām un ierobežojumiem, ir nepieciešams veikt to salīdzinājumu.

Pētījuma mērķis ir izpētīt un analizēt dažādas datubāžu pārvaldības sistēmas, lai noteiktu to piemērotību veidojam projektam.

Pētījuma uzdevumi:

1. Apkopot teorētisko informāciju par datubāžu pārvaldības sistēmu veidiem un to arhitektūrām.
2. Izvēlēties vairākas datu bāžu pārvaldības sistēmas analīzei.
3. Noteikt kritērijus, piemēram veiktspēja, lietošanas vienkāršība, mērogojamība, implementācija vienkāršība veidojamā sistēmā un atbalstītās funkcijas, kuri tiks ņemti vērā DBVS salīdzināšanā.
4. Veikt salīdzinošo analīzi, ņemot vērā noteiktos kritērijus.
5. Veikt secinājumus par to, kuras DBVS ir vispiemērotākās konkrētām projektam.

Studiju projekta pētījumā izmantotas šādas pētījuma metodes**:**

1. Literatūras un dokumentācijas izpēte
2. Datubāžu pārvaldības sistēmu salīdzinošā analīze
3. Datubāžu pārvaldības sistēmas aprobācija

## 1. Datubāžu pārvaldības sistēmas un to attīstības vēsture

1.1. Kas ir datubāžu pārvaldības sistēma

Datubāžu pārvaldības sistēma (DBVS) ir datorizēta datu glabāšanas sistēma. Sistēmas lietotājiem tiek nodrošinātas iespējas veikt dažādas darbības ar šo sistēmu, tās ir, datu manipulēšana un pašas datubāzes struktūras pārvaldīšana.

1.2. Datu bāžu vēsture

Datorizētas datubāzes aizsākumi sākas 1960.gados kad datori sāka kļūt pieejamāki privātuzņēmumiem un padarīja tos kā efektīvus risinājumus datu glabāšanai. Populārākie tā laika datu modeļi bija “CODASYL” un “IMS”.

Relāciju datubāžu aizsākumi sākās 1970.gados ar diviem datubāžu sistēmu prototipiem, tie ir, “Ingres”, izstrādāja uzņēmums “UBC”, un “System R”, izstrādāja uzņēmums “IBM”. “System R” izmanto vaicājumu valodu SEQUEL, mūsdienās zināms kā SQL, kura tiek plaši izmantota vēl mūsdienās. 1976.gadā Pīters Pin-Šans Čens ierosināja Entītiju-Relāciju jeb ER datubāžu modeli. Šis modelis ļauj izstrādātājiem koncentrēties uz datiem un to pielietojumu, nevis uz loģisko tabulu struktūru.

1980.gados “Structured Query Language” jeb SQL kļuva par plašu standarta vaicājumu valodu. Datoru pieejamības celšanās padarīja relāciju datubāžu sistēmas populārākas nekā iepriekš minētie datubāžu modeļi.

1990.gados ar interneta pirmsākumiem sāka palielināties datubāžu nepieciešamība. Klajā nāca vairākas datubāžu pārvaldības sistēmas, kuras vēljoprojām nav zaudējušas savu popularitāti mūsdienās, piemēram, “MySQL”, “Oracle Developer” u.c.

## 2. Datubāzu pārvaldības sistēmu salīdzinošā analīze

2.1. Vērtēšanas kritēriji

Izvērstie kritēriji un to pamatojums:

1. Datu apstrādes un meklēšanas ātrums

Šis kritērijs ir būtisks, jo tieši ietekmē lietotāja pieredzi un sistēmas veiktspēju. Lēna datu izgūšana vai apstrāde var radīt būtiskus kavējumus gan lietotnē, gan uzņēmuma procesos, tādā veidā lietotāji var zaudēt interesi lietot platformu.

1. Kompleksu datu tipu atbalsts

Projekta ietvaros ir paredzēts izmantot dažādus datu tipus ar dažādiem lielumiem. Dažādi datu tipi palīdz nodrošināt datu glabāšanu un optimizāciju.

1. Atbalsts transakcijām

Transakciju atbalsts nodrošina datu integritāti, tas ir, novērš negaidītas situācijas, piemēram, nejauša datu dzēšana vai maiņa.

1. Lietotāju piekļuves kontrole

Datubāzes pārvaldības sistēmai ir jāatbalsta lietotāju izveide un atļauju maiņu/noteikšanu. Šis kritērijs ir svarīgs, jo nesankcionēta pieeja datubāzei var būt bīstama, tas ir, lietotāju datu noplūde.

1. Saderība darbā ar Express.js veidojot API

Projekta izstrādē tiks veidots API ar Express.js, tādēļ ir nepieciešams pārliecināties ka programma var sazināties ar datubāzi.

1. Saderība ar Windows 10/11 vai Linux

Svarīgi, jo izstrādei neapšaubāmi tiks izmantots Windows 10/11 operētājsistēmas, tādēļ ir nepieciešams ka datubāžu pārvaldības sistēmas ir saderīgas.

1. Administrēšanas un konfigurācijas ērtums

Labi pārskatāmi rīki un saprotama konfigurācija ļauj ātri uzstādīt, pielāgot un uzturēt datubāzi bez nepieciešamības pēc dziļām tehniskām zināšanām.

1. Dokumentācijas un lietotāja interfeisa kvalitāte

Laba dokumentācija palīdz ātrāk apgūt un atrisināt problēmas. Viegli lietojams interfeiss palīdz nezaudēt laiku meklējot pareizās pogas un strādāt pie izstrādes.

1. Piedāvā vaicājumu automatizēšanu/ģenerēšanu

Ir iespējams veidot nejauši ģenerētu saturu. Tiek atvieglota testēšana un izstrāde ja nav nepieciešams manuāli veidot simtiem ierakstu.

**1. tabula vērtēšanas kritēriji**

**Kritēriju svars un punktu pamatojums**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Kritērijs | Svars | 3 punkti | 2 punkti | 1 punkts |
| 1.Lietošanas izmaksas | 20% | Bezmaksas visām vajadzībam | Bezmaksas ar ierobežojumiem balsoties uz uzņēmuma/produkta ieņēmumiem | Produkts nav pieejams bezmaksas |
| 2.Kompleksu datu tipu atbalsts | 10% | Pilnvērtīgs atbalsts kompleksiem datu tipiem (piemēram, ligzdoti JSON objekti, masīvi, ģeogrāfiskie dati, XML, binārie dati) ar iespēju tos efektīvi vaicāt un indeksēt. | Atbalsta pamata datu tipus, kā arī ierobežotu kompleksu datu tipu atbalstu (piemēram, JSON). Var būt nepieciešama manuāla apstrāde. | Atbalsta tikai pamata datu tipus (skaitļi, virknes, datumi, Būla vērtības). |
| 3.Atbalsts transakcijām | 5% | Pilnvērtīgs transakciju atbalsts ar visām īpašībām, ieskaitot sarežģītas, vairāku darbību transakcijas, ligzdotas transakcijas, un iespēju bloķēt resursus transakciju laikā. | Nodrošina pamata transakciju atbalstu ar iespēju veikt vienkāršas transakcijas (piemēram, INSERT, UPDATE, DELETE vienā tabulā). | Nav iebūvēta transakciju atbalsta. Datu integritātes nodrošināšanai nepieciešama manuāla loģika lietojumprogrammas līmenī. |
| 4.Lietotāju piekļuves kontrole | 5% | Lietotāju piekļuves kontrole ar detalizētām lomām un atļaujām (piemēram, piekļuve kolonnām, ierakstiem, skatiem), integrācija ar esošajām autentifikācijas sistēmām (piemēram OAuth). | Pamata lietotāju un lomu pārvaldība ar iespēju piešķirt pamata atļaujas (piemēram, lasīt, rakstīt, modificēt) datu bāzes vai tabulas. | Minimāla vai nekāda piekļuves kontrole. |
| 5.Saderība darbā ar Express.js veidojot API | 15% | Pastāv pirmās vai trešās puses bibliotēka. Bibliotēka ir kvalitatīvi dokumentēta un ir aktīvi lietots. | Pastāv trešās puses bibliotēka. Bibliotēkas nav kvalitatīvi dokumentētas un pastāv problēmas. | Nav tiešas integrācijas. |
| 6.Saderība ar Windows 10/11 vai Linux | 15% | Pilnīga un bezproblēmu saderība ar abām operētājsistēmām (Windows 10/11 un Linux). Viegla instalācija, vienāda funkcionalitāte un veiktspēja visās atbalstītajās vidēs. | Labs atbalsts abām operētājsistēmām (Windows 10/11 un Linux) ar visām pamatfunkcijām, taču var būt nelielas atšķirības instalācijā vai konfigurācijā. | Ierobežots atbalsts ar zināmām problēmām. |
| 7.Administrēšanas un konfigurācijas ērtums | 15% | Viegla un intuitīva administrēšana un konfigurācija, izmantojot labi izstrādātu lietotāja interfeisu un/vai viegli saprotamus konfigurācijas failus. Automatizēti rīki datu bāzes uzturēšanai, dublēšanai un atjaunošanai. | Pieejami pamata administrēšanas rīki (piemēram, komandrindas rīki, ierobežots grafiskais interfeiss), kas atvieglo ikdienas uzdevumus. Tomēr sarežģītākām konfigurācijām joprojām var būt nepieciešama manuāla iejaukšanās. | Sarežģīta administrēšana un konfigurācija, kas prasa dziļas tehniskās zināšanas un manuālu rediģēšanu konfigurācijas failos. Trūkst grafiskā interfeisa rīku. |
| 8.Dokumentācijas kvalitāte | 10% | Visaptveroša un regulāri atjaunināta dokumentācija ar detalizētiem piemēriem, apmācībām un labāko praksi. Aktīva kopiena un atbalsts. | Dokumentācija ar pamata informāciju, bet var trūkt detalizētu piemēru vai uzlabotu scenāriju aprakstu. | Nepilnīga, novecojusi vai grūti saprotama dokumentācija. Grūti atrast atbildes uz problēmām. |
| 9.Piedāvā vaicājumu automatizēšanu/ģenerēšanu | 5% | Sistēma piedāvā jaudīgus vaicājumu automatizācijas un ģenerēšanas rīkus, kas būtiski atvieglo datu apstrādi. Pastāv iespējas ģenerēt SQL vaicājumus no vizuālām shēmām vai iespējas definēt kodu kurš var nejauši izveidot vaicājumus, piemēram, kā tas ir iespējams PL/SQL. | Ir pieejamas ierobežotas vaicājumu automatizācijas iespējas, piemēram, vienkāršs vaicājumu veidotājs (Query Builder) ar pamatfunkcijām, vai koda snippets, kas palīdz samazināt manuālo darbu. | Sistēma nepiedāvā nekādas iebūvētas iespējas vaicājumu automatizēšanai vai ģenerēšanai. Viss jāraksta manuāli. |

2.2. Datubāzu pārvaldības sistēmas



PostgreSQL

PostgreSQL ir atvērtā koda objektu-relāciju datu bāzu sistēma, kas izmanto un paplašina SQL valodu, apvienojot to ar daudzām funkcijām, kas droši glabā un mērogo vissarežģītākās datu slodzes. PostgreSQL pirmsākumi sākās 1986. gadā kā daļa no POSTGRES projekta Kalifornijas Universitātē Bērklijā, un tās pamatplatformas aktīvā izstrāde notiek vairāk nekā 35 gadus.

PostgreSQL darbojas uz visām galvenajām operētājsistēmām, kopš 2001. gada ir ACID-saderīga, un tai ir jaudīgi papildinājumi, piemēram, populārais PostGIS ģeotelpiskās datu bāzes paplašinātājs.

1. PostgreSQL vērtējums

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1.Lietošanas izmaksas | *3 punkti* | *Produkts ir atvērtā koda programmatūra un bezmakasas* |
| 2.Kompleksu datu tipu atbalsts | *3 punkti* | *Pilnvērtīgs atbalsts kompleksiem datu tipiem (piemēram, ligzdoti JSON objekti, masīvi, ģeogrāfiskie dati, XML, binārie dati) ar iespēju tos efektīvi vaicāt un indeksēt.* |
| 3.Atbalsts transakcijām | *2 punkti* | *Transakcijas pastāv, bet nav ligzdotas transakcijas* |
| 4.Lietotāju piekļuves kontrole | *3 punkti* | Plaša lietotāju kontrole ar plašu klāstu autentifikācijas metožu. |
| 5.Saderība darbā ar Express.js veidojot API | *3 punkti* | *Pastāv trešās puses bibliotēkas ar plāšu dokumentāciju, aktīvi tiek izstrādāta un ar aktīvu kopienu.* |
| 6.Saderība ar Windows 10/11 vai Linux | *3 punkti* | *Ļoti plašs saderības sarkasts, iekļauj Windows un Linux operētājsistēmas.* |
| 7.Administrēšanas un konfigurācijas ērtums | *3 punkti* | *PostgreSQL nāk ar rīku pgadmin. Orientēties un atrasts nepieciešamās funckijās ir vienkārši un neskaidrību gadījumā informāciju ir viegli atrast forumos vai dokumentācijā.* |
| 8.Dokumentācijas kvalitāte | *3 punkti* | *Dokumentācija ir kvalitāti aprakstīta* |
| 9.Piedāvā vaicājumu automatizēšanu/ģenerēšanu | *3 punkti* | *Ir iespējams izmantot PL SQL programmēšanas valodu vaicājumu automatizēšānai* |

MySQL

MySQL ir atvērtā koda RDBMS, kas izmanto SQL, lai izveidotu un pārvaldītu datubāzes. Kā relāciju datubāze, MySQL glabā datus rindu un kolonnu tabulās, kas organizētas shēmās. Shēma definē, kā dati tiek organizēti un glabāti, un apraksta attiecības starp dažādām tabulām. Izmantojot šo formātu, izstrādātāji var viegli glabāt, izgūt un analizēt daudzus datu tipus, tostarp vienkāršu tekstu, skaitļus, datumus, laikus un, pavisam nesen, JSON un vektorus.

Tā kā MySQL ir atvērtā koda, tas ietver daudzas funkcijas, kas izstrādātas ciešā sadarbībā ar lietotāju kopienu gandrīz 30 gadu garumā. Divas iespējas, uz kurām paļaujas izstrādātāji, ir MySQL atbalsts ACID transakcijām un MySQL spēja mērogoties. Ar ACID transakcijām MySQL var garantēt, ka visas datu modifikācijas tiek veiktas konsekventi un uzticami, pat sistēmas avārijas gadījumā. MySQL var mērogot, lai atbalstītu ļoti lielas datubāzes, un tas var apstrādāt lielu vienlaicīgu savienojumu skaitu.

MySQL veiktspēja, lietošanas vienkāršība un zemās izmaksas, apvienojumā ar tā spēju uzticami mērogoties, augot biznesam, ir padarījušas to par pasaulē populārāko atvērtā koda datubāzi.

MySQL piederēja un to sponsorēja Zviedrijas uzņēmums MySQL AB, ko iegādājās Sun Microsystems (tagad Oracle Corporation). 2010. gadā, kad Oracle iegādājās Sun, Videnius atdalīja atvērtā koda MySQL projektu, lai izveidotu MariaDB.

1. MySQL vērtējums

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1.Lietošanas izmaksas | *2 punkti* | *MySQL ir Comunity Edition kuru var izmantot bezmakas, bet tas nāk ar mazāk rīkiem kā citas varsijas, Standart edition un Enterprise edition.* |
| 2.Kompleksu datu tipu atbalsts | *2 punkti* | *Lai gan tas atbalsta JSON un GIS, trūkst tieša masīvu un pilnvērtīga XML tipa atbalsta, kas nozīmē, ka sarežģītākos gadījumos var būt nepieciešama papildu manuāla apstrāde vai modelēšana.* |
| 3.Atbalsts transakcijām | *2 punkti* | *Transakcijas pastāv, bet nav ligzdotas transakcijas* |
| 4.Lietotāju piekļuves kontrole | *3 punkti* | *Augsta lietotāju kontrole, vairākas autentifikācijas metodes, atblasta autentifikācijas “plug-ins”* |
| 5.Saderība darbā ar Express.js veidojot API | *3 punkti* | *Pastāv trešās puses bibliotēkas ar plāšu dokumentāciju, aktīvi tiek izstrādāta un ar aktīvu kopienu.* |
| 6.Saderība ar Windows 10/11 vai Linux | *3 puntki* | *Ļoti plāšs OS atbalsts, Linux, Windows, macOS.* |
| 7.Administrēšanas un konfigurācijas ērtums | *2 punkti* | *Vienkārši izmantot, bet dažas funckijas var nestrādāt kā paredzēts, piemēram lietotāju administrēšāna var nestrādāt un ir nepieciešams izmantot komandas.* |
| 8.Dokumentācijas kvalitāte | *3 punkti* | *Dokumentācija ir kvalitāti aprakstīta* |
| 9.Piedāvā vaicājumu automatizēšanu/ģenerēšanu | *2 punkti* | *Iespējams automatizēt datubāzes izveidošanau pēc vizuālo shēmu zīmēšanas, bet nav iespējams automatizēt vaicājumus* |

MariaDB

MariaDB Server ir viens no populārākajiem datu bāzu serveriem pasaulē. To veidojuši MySQL oriģinālie izstrādātāji, un ir garantēts, ka tas paliks atvērtā koda. Ievērojami lietotāji ir Wikipedia, WordPress.com un Google.

MariaDB Server pārvērš datus strukturētā informācijā plašā lietojumprogrammu spektrā, sākot no banku darbības līdz tīmekļa vietnēm. Sākotnēji izstrādāts kā uzlabots, tieši aizvietojams MySQL, MariaDB Server tiek izmantots, jo tas ir ātrs, mērogojams un robusts, ar bagātīgu glabāšanas dzinēju, spraudņu un daudzu citu rīku ekosistēmu, kas padara to ļoti daudzpusīgu plašam lietošanas gadījumu klāstam.

MariaDB Server tiek izstrādāts kā atvērtā koda programmatūra, un kā relāciju datu bāze tas nodrošina SQL interfeisu datu piekļuvei. Jaunākās MariaDB Server versijas ietver arī ĢIS un JSON funkcijas.

1. MariaDB vērtējums

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1.Lietošanas izmaksas | *3 punkti* | *MariaDB tika izveidota ar mērķi saglabāt MySQL atvērto koda dabu, kad Oracle iegādājās MySQL. Produkts ir pilnībā bezmaksas.* |
| 2.Kompleksu datu tipu atbalsts | *2 puntki* | *Lai gan tas nodrošina labu JSON un GIS atbalstu jaunākajās versijās, tieša un visaptveroša masīvu un XML datu tipu integrācija datu bāzē, kāda ir redzama objektu-relāciju datubāzēs, nav pieejama.* |
| 3.Atbalsts transakcijām | *2 punkti* | *Transakcijas pastāv, bet nav ligzdotas transakcijas* |
| 4.Lietotāju piekļuves kontrole | *3 punkti* | *Augsta lietotāju kontrole, vairākas autentifikācijas metodes, atblasta autentifikācijas “plug-ins”* |
| 5.Saderība darbā ar Express.js veidojot API | *3 punkti* | *Pastāv trešās puses bibliotēkas ar plāšu dokumentāciju, aktīvi tiek izstrādāta un ar aktīvu kopienu.* |
| 6.Saderība ar Windows 10/11 vai Linux | *3 punkti* | *Operētājsistēmu saderība ietver pamatā tikai Linux un Windows operētājsistēmas, bet ieķlaujas 3 punktu vērtejumā.* |
| 7.Administrēšanas un konfigurācijas ērtums | *3 punkti* | *Vienkārši administrēt* |
| 8.Dokumentācijas kvalitāte | *3 punkti* | *Dokumentācija ir kvalitāti aprakstīta* |
| 9.Piedāvā vaicājumu automatizēšanu/ģenerēšanu | *2 punkti* | *Iespējams automatizēt datubāzes izveidošanau pēc vizuālo shēmu zīmēšanas, bet nav iespējams automatizēt vaicājumus* |

Microsoft SQL Server

Microsoft SQL Server ir relāciju datu bāzu pārvaldības sistēma, ko izstrādājis Microsoft, izmantojot strukturēto vaicājumu valodu SQL. Microsoft tirgo dažādu Microsoft SQL Server versiju, kas paredzētas dažādām auditorijām un darba slodzēm, sākot no mazām vienas mašīnas lietojumprogrammām līdz lielām, ar internetu savienotām lietojumprogrammām ar daudziem vienlaicīgiem lietotājiem.

1. Microsoft SQL Server vērtējums

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1.Lietošanas izmaksas | *1 punkts* | *Produkts ir pieejams bezmaksas apmācībām, bet datubāzes uzturēšanai ir nepieciešamas licences.* |
| 2.Kompleksu datu tipu atbalsts | *3 punkti* | *Microsoft SQL Server nodrošina pilnvērtīgu atbalstu daudziem kompleksiem datu tipiem, tostarp JSON, ģeogrāfiskajiem datiem un XML, ar iespēju tos efektīvi vaicāt un indeksēt.* |
| 3.Atbalsts transakcijām | *3 punkti* | *Pastāv transakcijas kopā ar atbalstu ligzdotām transakscijām.* |
| 4.Lietotāju piekļuves kontrole | *3 punkti* | *Augsta lietotāju kontrole, vairākas autentifikācijas metodes, atblasta autentifikācijas “plug-ins”* |
| 5.Saderība darbā ar Express.js veidojot API | *3 punkti* | *Pastāv trešās puses bibliotēkas ar plāšu dokumentāciju, aktīvi tiek izstrādāta un ar aktīvu kopienu.* |
| 6.Saderība ar Windows 10/11 vai Linux | *2 punkti* | *Saderība tikai Windows operētājs sistēamām (Windows 10 vai jaunāks, iekļauj windows server versijas, tās ir, Windows Server 2016 un jaunaks)* |
| 7.Administrēšanas un konfigurācijas ērtums | *2 punkti* | *Nedaudz sarežģīti atrast funkcijas un logus, bet bez dokumentācijas nav pārāk lielas problēmas.* |
| 8.Dokumentācijas kvalitāte | *3 punkti* | *Dokumentācija ir kvalitāti aprakstīta* |
| 9.Piedāvā vaicājumu automatizēšanu/ģenerēšanu | *3 punkti* | *Ir iespējams izmantot T-SQL programmēšanas valodu vaicājumu automatizēšānai* |

Oracle Developer

Oracle SQL Developer ir bezmaksas, integrēta izstrādes vide, kas vienkāršo Oracle datubāzes izstrādi un pārvaldību gan tradicionālajās, gan mākoņa izvietošanās. SQL Developer piedāvā pilnīgu PL/SQL lietojumprogrammu izstrādi no sākuma līdz beigām, darblapu vaicājumu un skriptu palaišanai, atskaišu saskarni, pilnīgu datu modelēšanas risinājumu un migrācijas platformu trešo pušu datubāzu pārvietošanai uz Oracle.

1. Oracle Developer vērtējums

|  |  |  |
| --- | --- | --- |
| **Kritērijs** | **Novērtējums** | **Pamatojums** |
| 1.Lietošanas izmaksas | *1 punkts* | *Ir bezmakasas iespējas, bet ļoti limitēts izmēros un pastāv problēmas pieejamību.* |
| 2.Kompleksu datu tipu atbalsts | *3 punkti* | *Oracle Database ir viena no spēcīgākajām datubāzēm kompleksu datu tipu atbalsta ziņā (JSON, XML, telpiskie dati, kolekcijas utt.).* |
| 3.Atbalsts transakcijām | *3 punkti* | *Pastāv transakcijas kopā ar atbalstu ligzdotām transakscijām.* |
| 4.Lietotāju piekļuves kontrole | *3 punkti* | *Augsta lietotāju kontrole, vairākas autentifikācijas metodes, atblasta autentifikācijas “plug-ins”* |
| 5.Saderība darbā ar Express.js veidojot API | *3 punkti* | *Pastāv trešās puses bibliotēkas ar plāšu dokumentāciju, aktīvi tiek izstrādāta un ar aktīvu kopienu.* |
| 6.Saderība ar Windows 10/11 vai Linux | *2 punkti* | *Tikai Windows operētājsitēmas (No Windows Vista līdz Windows 11)* |
| 7.Administrēšanas un konfigurācijas ērtums | *1 punkti* | *Sarežģīti iesākt izmantot, jo visas funckijas stāv vienā sarkastā un to ir ļoti daudz, dokumentācijas vai forumu izmantošana ir nepieciešama.* |
| 8.Dokumentācijas kvalitāte | *3 punkti* | *Dokumentācija ir kvalitāti aprakstīta* |
| 9.Piedāvā vaicājumu automatizēšanu/ģenerēšanu | *3 punkti* | *Ir iespējams izmantot PL SQL programmēšanas valodu vaicājumu automatizēšānai* |

## 2.3. Datubāzu pārvaldības sistēmu salīdzinošā analīze

1. DBPS vērtējumu pārskats

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **DBPS** | 1.Lietošanas izmaksas | 2.Kompleksu datu tipu atbalsts | 3.Atbalsts transakcijām | 4.Lietotāju piekļuves kontrole | 5.Saderība darbā ar Express.js veidojot API | 6.Saderība ar Windows 10/11 vai Linux | 7.Administrēšanas un konfigurācijas ērtums | 8.Dokumentācijas kvalitāte | 9.Piedāvā vaicājumu automatizēšanu/ģenerēšanu | **Vērtējums** | |
| Svars | 20% | 10% | 5% | 5% | 15% | 15% | 15% | 10% | 5% | 100% |
| *PostgreSQL* | 20 | 10 | 3.33 | 5 | 15 | 15 | 15 | 10 | 5 | 98.33  % |
| *MySQL* | 13.33 | 6.67 | 3.33 | 5 | 15 | 15 | 10 | 10 | 3.33 | 81.66% |
| *MariaDB* | 20 | 6.67 | 3.33 | 5 | 15 | 15 | 15 | 10 | 3.33 | 93.33% |
| *Microsoft SQL Server* | 6.67 | 10 | 5 | 5 | 15 | 10 | 10 | 10 | 5 | 76.67% |
| *Oracle Developer* | 6.67 | 10 | 5 | 5 | 15 | 10 | 5 | 10 | 5 | 71.67% |

1.tabulas rindiņā „Svars” ierakstīti katra kritērija būtiskuma skaitlisks vērtējums atbilstoši “Code Moodle” projekta prasībām. *.* Svara koeficientu vērtējums ir subjektīvi noteicis darba autors. *Kā redzams pēc analīzes tabulas datiem, “Code Moodle” projekta izstrādei vispiemērotākā ir PostgreSQL, kas vērtējumā ir ieguvis augstāko rezultātu 98.33%.*

## 2.4. Analīzes rezultāti

Secinājumi

1. Pētījumā PostgreSQL ieguva visaugstāko kopējo vērtējumu. Tā izceļas ar:

* Plašu funkcionalitāti, tostarp pilnvērtīgu kompleksu datu tipu atbalstu (JSON, masīvi, XML, GIS u.c.),
* Stabilu transakciju atbalstu un detalizētu piekļuves kontroli,
* Pilnu savietojamību ar Express.js un abām populārākajām operētājsistēmām (Windows un Linux),
* Ērtu administrēšanu ar rīku “pgAdmin” un kvalitatīvu dokumentāciju,
* Iespēju izmantot vaicājumu ģenerēšanas un automatizācijas rīkus, piemēram, PL/pgSQL.

Šīs īpašības padara PostgreSQL īpaši piemērotu projektam, kur nepieciešama elastība, mērogojamība, drošība un strādāšana ar sarežģītiem datiem.

2. MySQL ir piemērots, bet nāk ar ierobežojumiem salīdzinot ar PostgreSQL. Projekta ietveros MySQL ir derīgs, jo ir pastāvošas zināšanas un noteiktās imitācijas neietekmētu projekta izstrādi, ja projekts paliek tikai studiju projekta ietvaros.

## Jēdzienu skaidrojumi un saīsinājumi

* DBVS – Datubāžu pārvaldības sistēma
* ACID – Princips kurš apzīmē "atomitāti, konsekvenci, izolāciju un izturību" (atomicity, consistency, isolation, and durability).
* OS – Operētājsistēma

## Izmantotā literatūra

1. *Quickbase Pieejams: https://www.quickbase.com/articles/timeline-of-database-history* *[skatīts 13.05.2025]*
2. *IBM Pieejams: https://www.ibm.com/docs/en/zos-basic-skills?topic=zos-what-is-database-management-system [skatīts 13.05.2025]*
3. *GeeksForGeeks Pieejams: https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/ [skatīts 13.05.2025]*
4. *Wikipedia Pieejams: https://en.wikipedia.org/wiki/Peter\_Chen [skatīts 13.05.2025]*
5. *Wikipedia Pieejams: https://en.wikipedia.org/wiki/SQL [skatīts 13.05.2025]*
6. AlexSoft Pieejams: *https://www.altexsoft.com/blog/comparing-database-management-systems-mysql-postgresql-mssql-server-mongodb-elasticsearch-and-others/ [skatīts 13.05.2025]*
7. Medium Pieejams: *https://medium.com/wix-engineering/how-to-choose-the-right-database-for-your-service-97b1670c5632 [skatīts 13.05.2025]*
8. Intergrate Pieejams: *https://www.integrate.io/blog/which-database/ [skatīts 13.05.2025]*
9. *Oracle Pieejams: https://www.oracle.com/mysql/what-is-mysql/ [skatīts 13.05.2025]*
10. *Wikipedia Pieejams: https://lv.wikipedia.org/wiki/MySQL* *[skatīts 13.05.2025]*
11. *Wikipedia Pieejams: https://en.wikipedia.org/wiki/Oracle\_SQL\_Developer [skatīts 13.05.2025]*
12. *Wikipedia Pieejams: https://en.wikipedia.org/wiki/Microsoft\_SQL\_Server [skatīts 13.05.2025]*
13. *SQLite Pieejams: https://www.sqlite.org/about.html [skatīts 13.05.2025]*
14. *PostgreSQL Pieejams: https://www.postgresql.org/about/ [skatīts 13.05.2025]*
15. *MariaDB Pieejams: https://mariadb.org/about/ [skatīts 13.05.2025]*
16. *OpenJS Foundation Pieejams: https://expressjs.com/en/guide/database-integration.html [skatīts 13.05.2025]*
17. *MariaDB Pieejams: https://mariadb.com/kb/en/getting-started-with-the-node-js- connector/ [skatīts 13.05.2025]*

# II daļa <(programmatūras projekts, komandas veikums)>

## 1. “Code Moodle” projekta apraksts

1.1 Projekta mērķis un uzdevumi

Projekta mērķis ir izveidot sistēmu, kura atvieglo studentiem un skolotājiem kodēšanas uzdevumu uzdošanu, veikšanu un vērtēšanu. Projekta uzdevumi ir izveidot datubāzi, datubāzes saskarni un izveidot klienta pusi. Datubāzei ir jāspēj glabāt visus nepieciešamos datus, no lietotāju ielogošanās datiem, līdz kodēšanas uzdevumu iesniegumiem. Datubāzes saskarne ir nepieciešama lai lietotājiem nebūtu tieša pieeja datubāzes savienojumam, procesi varētu tikt automatizēti, piemēram, reģistrācija, klienta pusē pārbauda vai ievaddati ir derīgi, tad datubāzes saskarne pārbauda vai lietotājs ar doto epastu jau nepastāv un ja nē tad izveido lietotāju. Klienta puse ir saskarne kur lietotāji spēs apmeklēt un lietot, klienta puse veic datu vizualizāciju balstoties uz lietotāja lomām, piemēram, skolotāji spēj veidot uzdevumus un moduļus kur skolēni redz tikai pieejamos moduļus, uzdevumus un iesniegumu vēsturi. Klienta puse arī veic datu validāciju, tas ir, vai dati, tādi kā epasts vai parole ir derīgi. Datu validācija neietver datu duplikācijas pārbaudes, to veic datubāzes saskarne.

* 1. Projekta organizatoriskā struktūra

Komandas sastāvs:

* Kārlis Lācītis
* Endijs Dārznieks

1. Komandas lomu sadalījums, uzdevumi un prasības

|  |  |  |
| --- | --- | --- |
| Lomas nosaukums | **Dalībnieki** | **Lomas darba uzdevumi** |
| Projekta vadītājs | Kārlis Lācītis | Darba organizācija |
| Sistēmas saskarnes izstrādātājs | Kārlis Lācītis, Endijs Dārznieks | Sistēmas klienta puses izstrāde |
| Sistēmas servera izstrādātājs | Kārlis Lācītis, Endijs Dārznieks | Sistēmas API izstrāde |
| Dizaineris | Endijs Dārznieks | Sistēmas klienta puses izstrāde |
| Analists | Endijs Dārznieks | Prasību noteikšana |
| Datubāzes modelētājs | Kārlis Lācītis | Datubāzes modelēšana |
| Testētājs | Kārlis Lācītis, Endijs Dārznieks | Testēšanas žurnāla sagatave, LRG izstrāde |
| Arhitekts | Endijs Dārznieks | Projekta publiskās hostēšanas nodrošināšana |

Komandas komunikācijas rīki:

* Sarunas klātienē
* Discord
* Trello

1.3 Projekta nodevumi

Projekta nodevumi:

1. PPS
2. PPA
3. PPPP
4. Intervijas protokols
5. Testēšanas žurnāls
6. LRG
7. Individuālais pētījums / darba apraksts.
8. Klienta puses pirmkods
9. Servera puses pirmkods

Visi nodevumi ir pieejami <https://github.com/EndijsD/code-moodle>

## 2. Projekta nodevumu apraksts

Nodevumi iedalāmi trīs grupās(kopīgs):

* Programmatūras projekta pārvaldības plāns (PPPP);
* programmatūras dokumentācija:
  + Programmatūras prasību specifikācija (PPS), Programmatūras projektējuma apraksts (PPA), lietotāja rokasgrāmata (LRG),
  + testēšanas dokumentācija (TST) (testpiemēri, testēšanas žurnāls, Problēmziņojumi un Problēmu/kļūdu ziņojumu reģistrācijas žurnāls, testēšanas kopsavilkums);
* pirmkodi – Pirmkods tiek iedalīts divās daļās :
  + Klienta puses pirmkods – satur pirmkodu mājaslapas struktūrai, datu vizualizācijai un vaicājumu izsaukšanai un projekta servera pusi.
  + Servera puses pirmkods – pirmkods apstrādā vaicājumus no klienta puses, veic pārbaudes un veic datu manipulācijas datubāzē
* Datubāze un tās shēma – nodevumā nāks MySQL datubāzes shēmas fails, kur ir iespējams apskatīt datubāzes ER shēmu. Ar rīku forward engineering var automatizēt vaicājumu izvedi datubāzes izveidei. Kopā ar shēmu nāk kopā .sql fails kurš satur datubāzes izveidi kopā ar sākumierakstiem testēšanai.

## 3. Autora un pārējo komandas dalībnieku ieguldījums projekta tapšanā

*<Projekta komandas kopīgais raksturojums. Apraksta katra komandas dalībnieka lomu programmatūras izstrādes komandā, veiktos uzdevumus, un sasniegtos rezultātus- komandas dalībnieku ieguldījums projekta izstrādē****,*** *kopējais procentuālais ieguldījus komandas darbā (aizpildīt tabulā 4).Kopīgs>*

***<Autora******ieguldījums projekta izstrādē*** *–* ***galvenie pienākumi, ieguldījus komandas darbā, radītie nodevumi. INDIVIDUĀLI>***

**4.tabula**

**Kopējais procentuālais ieguldījus komandas darbā**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dalībnieks** | **Programmatūra** | | | | | | **Pārvaldība** | **Kopā** |
| **PPS** | **PPA** | **LRG** | **TST** | **Kods** | **DB** | **PPPP** |
| Kārlis Lācīts | 50% | 60% | 40% | 40% | 50% | 50% | 70% | **360%** |
| Endijs Dārznieks | 50% | 40% | 60% | 60% | 50% | 50% | 30% | **340%** |
|  | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **100%** | **700%** |

**Darba vērtējums**

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi un manā darbā nav pārkāpts likums par autortiesībām.

Darba autors: Kārlis Lācītis ........................ \_\_.06.2025.

Studiju darbs aizstāvēts Liepājas Universitātes

Dabas un inženierzinātņu fakultātes

studiju darbu aizstāvēšanas komisijas sēdē ........................................................

(datums)

Novērtējums ...................................................

studiju darba vadītājs: Dzintars Tomsons.........................................