**JELGAVAS TEHNIKUMS**

**ERNESTS BALODIS**  
**WEB spēles prototipa izveide “B.Iespējas”**  
**Kvalifikācijas darbs kvalifikācijas ieguvei programmēšanas tehniķis**

Darba izpildītājs: 410. gr. izgl. E. Balodis  
*paraksts, datums*

**Jelgava 2025**

**ANOTĀCIJA**

**Balodis E. WEB spēles izstrāde “B.Iespējas” projektam: kvalifikācijas darbs. Jelgava: Jt, 2024. 109 lpp., 48 att., 13 tab., 22 pielikumi.**

Projekts **"b.iespejas"** ir izstrādāts kā interaktīva, uz pārlūkprogrammu balstīta spēle ar radošām un izglītojošām iespējām, kas sniedz lietotājiem iespēju veidot, kombinēt un saglabāt dažādus elementus. Spēles galvenais mērķis ir veicināt lietotāja iztēli un loģisko domāšanu, apvienojot dažādus elementus, lai radītu jaunus, loģiskus vai radošus rezultātus.

Spēle piedāvā dinamisku lietotāja saskarni, kurā spēlētāji var brīvi vilkt un nomest elementus, savienojot tos, lai izveidotu jaunus objektus. Katrs kombinācijas rezultāts tiek ģenerēts, izmantojot pielāgotu algoritmu, kas ir balstīts uz loģiskām un dabiskām likumsakarībām, un atspoguļo rezultātu ar simboliem un emocijzīmēm. Spēles ietvaros tiek izmantoti četri pamatelementi – uguns, ūdens, zeme un gaiss –, kurus spēlētājs var kombinēt, lai radītu sarežģītākus un unikālākus elementus.

Lietotājiem ir iespēja saglabāt izveidotos elementus savā profilā, veidojot personalizētu bibliotēku ar unikālām kombinācijām. Saglabātie elementi tiek glabāti datubāzē, kas nodrošina spēles progresu un ļauj lietotājiem turpināt savu spēles pieredzi jebkurā laikā. Spēle ir izstrādāta, izmantojot *PHP* kā servera puses valodu, *JavaScript* lietotāja saskarnes funkcionalitātei un *MariaDB* datubāzi lietotāja un elementu datu pārvaldībai.

***ANNOTATION***

**Balodis E. *Development* *of* *a WEB* *Game for the* “B.Iespējas” *Project*: *Qualification Work*. Jelgava: Jt, 2024. 109 *pages*, 48 *figures*, 13 *tables*, 22 *appendices*.**

*The "b.iespejas" project is developed as an interactive, browser-based game with creative and educational features, providing users with the opportunity to create, combine, and save various elements. The main objective of the game is to stimulate the user’s imagination and logical thinking by allowing them to combine different elements to produce new, logical, or creative results.*

*The game offers a dynamic user interface where players can freely drag and drop elements, combining them to create new objects. Each combination result is generated using a custom algorithm based on logical and natural patterns, and the outcome is represented by symbols and emojis. The game starts with four basic elements – fire, water, earth, and air – which players can combine to form more complex and unique elements.*

*Users have the ability to save the created elements in their profile, building a personalized library of unique combinations. The saved elements are stored in a database, ensuring game progress is preserved and allowing users to continue their experience at any time. The game is developed using PHP as the server-side language, JavaScript for the user interface functionality, and MariaDB for managing user and element data.*

**SATURS**

[IEVADS 9](#_Toc200052184)

[Akronīmi un saīsinājumi 10](#_Toc200052185)

[1. Uzdevuma Nostādne 11](#_Toc200052186)

[1.1. Līdzīgu programmatūru analīze 11](#_Toc200052187)

[1.1.1. Tiešsaistes spēle “Little Alchemy 2” 11](#_Toc200052188)

[1.1.2. Tiešsaistes spēle “Atomas” 12](#_Toc200052189)

[1.1.3. Tiešsaistes spēle “Doodle God” 14](#_Toc200052190)

[2. Prasību specifikācija 16](#_Toc200052191)

[2.1. Ievads 16](#_Toc200052192)

[2.1.1. Dokumenta nolūks 16](#_Toc200052193)

[2.1.2. Darbības sfēra 16](#_Toc200052194)

[2.1.3. Definīcijas 16](#_Toc200052195)

[2.1.4. Dokumenta pārskats 17](#_Toc200052196)

[2.2. Vispārīgs apraksts 17](#_Toc200052197)

[2.2.1. Spēles perspektīva 17](#_Toc200052198)

[2.2.2. Spēles funkcijas 17](#_Toc200052199)

[2.2.3. Lietotāja raksturiezīmes 18](#_Toc200052200)

[2.2.4. Spēles līmeņi 18](#_Toc200052201)

[2.2.5. darbības apraksts 19](#_Toc200052202)

[2.2.6. Ierobežojumi 19](#_Toc200052203)

[2.2.7. Klienta prasības 19](#_Toc200052204)

[2.3. Funkcionālais un dinamiskais sistēmas modelis 21](#_Toc200052205)

[2.3.1. Lietojumgadījumu diagramma 21](#_Toc200052206)

[2.4. Funkcionālās prasības 22](#_Toc200052207)

[2.4.1. Lietotāja reģistrācija 22](#_Toc200052208)

[2.4.2. Lietotāja pieteikšanās 24](#_Toc200052209)

[2.4.3. Spēles elementu kombinācijas un rezultātu ģenerēšana 26](#_Toc200052210)

[2.4.4. Elementu saglabāšana 28](#_Toc200052211)

[2.4.5. Elementu meklēšanas funkcija 29](#_Toc200052212)

[2.5. Nefunkcionālās prasības 30](#_Toc200052213)

[2.5.1. Kopējās nefunkcionālās prasības 30](#_Toc200052214)

[2.5.1.1. Standarta poga 30](#_Toc200052215)

[2.5.1.2. Satura virsraksti 30](#_Toc200052216)

[2.5.1.3. Ievadlauki 31](#_Toc200052217)

[2.5.1.4. Paroles parametra izskats ievades laikā 31](#_Toc200052218)

[2.5.1.5. Ievadlauku nosaukumi 32](#_Toc200052219)

[2.5.1.6. Elements 32](#_Toc200052220)

[2.5.2. Reģistrācijas logs 33](#_Toc200052221)

[2.5.3. Pieteikšanās logs 34](#_Toc200052222)

[2.5.4. Sākumlapas logs 35](#_Toc200052223)

[2.5.5. Spēles logs 35](#_Toc200052224)

[2.5.6. Biblotekas logs 37](#_Toc200052225)

[3. uzdevumu risināšanas līdzekļu pamantojums 39](#_Toc200052226)

[3.1. Koda redaktors 39](#_Toc200052227)

[3.2. Mākslīgais intelekts 39](#_Toc200052228)

[3.3. Datubāzes projektēšanas rīks 39](#_Toc200052229)

[3.4. Lokālā servera izveides rīks 40](#_Toc200052230)

[3.5. Blokshēmu izveides rīks 40](#_Toc200052231)

[3.6. Programmēšanas valodas 40](#_Toc200052232)

[3.7. *Wireframe* izveides rīks 41](#_Toc200052233)

[4. Datu struktūras apraksts 42](#_Toc200052234)

[4.1. Programmēšanas valodas: 42](#_Toc200052235)

[4.2. Datu bāzes pārvaldības sistēmas: 42](#_Toc200052236)

[4.3. Izstrādes vide: 42](#_Toc200052237)

[4.4. Papildrīki: 42](#_Toc200052238)

[4.5. Datu plūsmas diagrammas 43](#_Toc200052239)

[4.6. Stāvokļa diagramma 44](#_Toc200052240)

[4.7. ER modelis 44](#_Toc200052241)

[4.8. Sistēmas struktūras modelis 45](#_Toc200052242)

[4.8.1. Datubāzes apraksts 45](#_Toc200052243)

[4.8.2. “blueprints” tabula 45](#_Toc200052244)

[4.8.3. “saved\_elements” tabula 46](#_Toc200052245)

[4.8.4. “users” tabula 46](#_Toc200052246)

[4.9. Scenārijs – elementa reģistrācija 46](#_Toc200052247)

[4.10. Scenārijs – resursu meklēšana bibloteka 47](#_Toc200052248)

[4.11. Scenārijs – jauna spēlētāja reģistrācija 48](#_Toc200052249)

[4.12. Datu plūsmas process sistēmā 48](#_Toc200052250)

[5. testēšana 49](#_Toc200052251)

[5.1. Reģistrēšanās funkcijas testēšana 49](#_Toc200052252)

[5.2. Pieteikšanās funkcijas testēšana 49](#_Toc200052253)

[5.3. Resursu meklēšanas funkcijas testēšana 50](#_Toc200052254)

[5.4. Objekta veidošanas funkcijas testēšana 50](#_Toc200052255)

[5.5. Elementa notīrīšanas funkcijas testēšana 51](#_Toc200052256)

[5.6. Visu elementa notīrīšanas funkcijas testēšana 51](#_Toc200052257)

[5.7. Elementa saglabāšanas funkcijas testēšana 51](#_Toc200052258)

[5.8. Elementa atrašanās biblotekā testēšana 52](#_Toc200052259)

[5.9. Atgriežanās funkcijas testēšana 52](#_Toc200052260)

[5.10. Izrakstīšanās funkcijas testēšana 52](#_Toc200052261)

[6. Lietotāja ceļvedis 53](#_Toc200052262)

[6.1. Ievads 53](#_Toc200052263)

[6.2. Produkta pieejamība (prototipa stadijā) 53](#_Toc200052264)

[6.3. Pieteikšanās logs 53](#_Toc200052265)

[6.3.1. Vispārējs apraksts 53](#_Toc200052266)

[6.3.2. Darbība 54](#_Toc200052267)

[6.4. Reģistrēšanās lapa 56](#_Toc200052268)

[6.4.1. Vispārējs apraksts 56](#_Toc200052269)

[6.4.2. Darbība 57](#_Toc200052270)

[6.5. Sākumlapa 59](#_Toc200052271)

[6.5.1. Vispārējs apraksts 59](#_Toc200052272)

[6.5.2. Darbība 60](#_Toc200052273)

[6.6. Bibliotēkas logs 61](#_Toc200052274)

[6.6.1. Vispārējs apraksts 61](#_Toc200052275)

[6.6.2. Darbība 61](#_Toc200052276)

[6.7. Spēles logs 64](#_Toc200052277)

[6.7.1. Vispārējs apraksts 64](#_Toc200052278)

[6.7.2. Darbība 65](#_Toc200052279)

[Secinājumi un priekšlikumi 73](#_Toc200052280)

[Izmantoto avotu saraksts 75](#_Toc200052281)

[1. pielikums **game.js** 77](#_Toc200052282)

[2. pielikums **homepage.js** 82](#_Toc200052283)

[3. pielikums **index.js** 83](#_Toc200052284)

[4. pielikums **library.js** 84](#_Toc200052285)

[5. pielikums **game.css** 85](#_Toc200052286)

[6. pielikums **homepage.css** 88](#_Toc200052287)

[7. pielikums **index.css** 90](#_Toc200052288)

[8. pielikums **library.css** 93](#_Toc200052289)

[9. pielikums **combine\_elements.php** 95](#_Toc200052290)

[10. pielikums **db.php** 97](#_Toc200052291)

[11. pielikums **fetch\_elements.php** 98](#_Toc200052292)

[12. pielikums **fetch\_library\_elements.php** 99](#_Toc200052293)

[13. pielikums **fetch\_saved\_elements.php** 100](#_Toc200052294)

[14. pielikums **fetch\_saved\_game.php** 101](#_Toc200052295)

[15. pielikums **game.html** 102](#_Toc200052296)

[16. pielikums **homepage.html** 103](#_Toc200052297)

[17. pielikums **index.html** 104](#_Toc200052298)

[18. pielikums **library.html** 106](#_Toc200052299)

[19. pielikums **login.php** 107](#_Toc200052300)

[20. pielikums **logout.php** 108](#_Toc200052301)

[21. pielikums **register.php** 109](#_Toc200052302)

[22. pielikums **save\_element.php** 110](#_Toc200052303)

**ATTĒLU SARAKSTS**

[1. att. “*Little Alchemy 2*” 12](#_Toc200052304)

[2. att. “*Little Alchemy 2*” Reģistrēšanās lapa 12](#_Toc200052305)

[3. att. “*Atomas*” 13](#_Toc200052306)

[4. att. “*Atomas*” Pamat elementi 14](#_Toc200052307)

[5. att. “*Doodle* *God*” Izaicinājumi 14](#_Toc200052308)

[6. att. Lietojumgadījumu diagramma 21](#_Toc200052309)

[7. att. Reģistrēšanās funkcijas blokshēma 23](#_Toc200052310)

[8. att. Pieteikšanās funkcijas blokshēma 25](#_Toc200052311)

[9. att. kombinācijas funkcijas blokshēma 28](#_Toc200052312)

[10. att. Jaunu elementu funkcijas blokshēma 29](#_Toc200052313)

[11. att. kombinācijas funkcijas blokshēma 29](#_Toc200052314)

[12. att. Pogas pirms pārbraukšanas ar kursoru 30](#_Toc200052315)

[13. att. Poga pēc pārbraukšanas ar kursoru 30](#_Toc200052316)

[14. att. Loga satura virsraksta makets 31](#_Toc200052317)

[15. att. Ievadlauks pirms tajā ieklikšķināts 31](#_Toc200052318)

[16. att. Ievadlauka, ja tajā raksta paroli 31](#_Toc200052319)

[17. att. Ja prasības izspildīts: 32](#_Toc200052320)

[18. att. Ja prasības nav izpildītas: 32](#_Toc200052321)

[19. att. Ievadlauka nosaukums 32](#_Toc200052322)

[20. att. Elements 33](#_Toc200052323)

[21. att. Reģistrācijas logs 33](#_Toc200052324)

[22. att. Pieteikšanās logs 35](#_Toc200052325)

[23. att. Sākumlapas logs 35](#_Toc200052326)

[24. att. Spēles logs 37](#_Toc200052327)

[25. att. Biblotekas logs 38](#_Toc200052328)

[26. att. Lietotāja autentifikācijas datu plūsma 43](#_Toc200052329)

[27. att. Elementu reģistrācijas datu plūsma 43](#_Toc200052330)

[28. att. Meklēšanas un pārvaldības datu plūsma 44](#_Toc200052331)

[29. att. Entitāšu diagramma [Autora veidota ekrānkopija] 45](#_Toc200052332)

[30. att. Neaizpildītas pieteikšanās lapas izskats. 54](#_Toc200052333)

[31. att. Aizpildītas pierakstīšanās lapas izskats. 55](#_Toc200052334)

[32. att. Pāreja uz reģistrēšanās lapu. 56](#_Toc200052335)

[33. att. Neaizpildītas reģistrēšanās lapas izskats. 57](#_Toc200052336)

[34. att. Aizpildītas reģistrēšanās lapas izskats. 58](#_Toc200052337)

[35. att. Pāreja uz pierakstīšanās lapu. 59](#_Toc200052338)

[36. att. Sākumlapas izskats ar pieejamajām pogām. 60](#_Toc200052339)

[37. att. Bibliotēkas lapas izskats bez elementiem. 61](#_Toc200052340)

[38. att. Biblotekas loga elementu saraksts 62](#_Toc200052341)

[39. att. Meklēšanas josla darbībā 63](#_Toc200052342)

[40. att. Atgriezsties poga biblotekas logā 64](#_Toc200052343)

[41. att. Spēles loga izskats. 65](#_Toc200052344)

[42. att. Elements spēles laukumā 66](#_Toc200052345)

[43. att. Jauns elements spēles laukumā 67](#_Toc200052346)

[44. att. Saglabāt šeit lauks 68](#_Toc200052347)

[45. att. Meklēšanas josla darbībā elementu joslā 69](#_Toc200052348)

[46. att. Notīrīt poga 70](#_Toc200052349)

[47. att. Notīrīt šeit lauks 71](#_Toc200052350)

[48. att. Atgriezties poga spēles loga 72](#_Toc200052351)

**TABULU SARAKSTS**

[1. Tabula “*elements*” 45](#_Toc200052352)

[2. Tabula “*saved\_elements*” 46](#_Toc200052353)

[3. Tabula “*users*” 46](#_Toc200052354)

[4. Tabula reģistrēšanās funkcijas testēšana 49](#_Toc200052355)

[5. Tabula pieteikšanās funkcijas testēšana 50](#_Toc200052356)

[6. Tabula resursu meklēšanas funkcijas testēšana 50](#_Toc200052357)

[7. Tabula objekta veidošanas funkcijas testēšana 51](#_Toc200052358)

[8. Tabula Elementa notīrīšanas funkcijas testēšana 51](#_Toc200052359)

[9. Tabula Visu elementa notīrīšanas funkcijas testēšana 51](#_Toc200052360)

[10. Tabula Elementa saglabāšanas funkcijas testēšana 51](#_Toc200052361)

[11. Tabula Elementa atrašanās biblotekā testēšana 52](#_Toc200052362)

[12. Tabula Atgriežanās funkcijas testēšana 52](#_Toc200052363)

[13. Tabula Izrakstīšanās funkcijas testēšana 52](#_Toc200052364)

IEVADS

Šis dokuments sniedz detalizētu programmatūras prasību specifikāciju spēlei, kurā lietotāji apvieno dažādus elementus (ūdeni, uguni, zemi un gaisu), lai radītu jaunus priekšmetus un radības. Spēles pamatā ir smilšu kastes struktūra, kas nozīmē, ka lietotāji var eksperimentēt bez noteiktiem mērķiem. Mākslīgā intelekta integrācija nodrošina gandrīz neierobežotu kombināciju iespējas, ļaujot spēlētājiem izzināt un radīt savus atklājumus.

Dokumentā tiks izklāstīti programmatūras tehniskie aspekti, funkcionalitāte, lietotāja prasības un sistēmas darbības principi.

**Darba mērķis:**

Izveidot spēles prototipu, kurā lietotājiem būs iespēja reģistrēties, apvienot elementus, radīt jaunus priekšmetus un radības, kā arī saglabāt savus atklājumus turpmākajai izmantošanai. Mākslīgais intelekts palīdzēs ģenerēt jaunas kombinācijas, paplašinot spēlētāja radošās iespējas.

**Darba uzdevumi:**

1. Noskaidrot projekta prasības un izpētīt līdzīgu spēļu platformas, lai saprastu to kombināciju mehāniku un lietotāja interfeisu.
2. Analizēt mākslīgā intelekta algoritmus un to piemērošanu spēlē, lai nodrošinātu dinamiski ģenerētas kombinācijas.
3. Izstrādāt spēles dizainu, ņemot vērā lietotāju pieredzi un spēļu industrijas tendences.
4. Veidot funkcionalitāti, kas atbalsta pamatelementu apvienošanu un nodrošina iespēju saglabāt un izmantot izveidotās kombinācijas.
5. Izveidot spēlētāju radošuma attīstīšanai piemērotu smilšu kastes struktūru bez konkrētiem spēles mērķiem.
6. Nodrošināt, ka lietotāju radītie dati un kombinācijas tiek droši glabāti un viegli pieejami turpmākai izmantošanai.
7. Testēt izveidoto spēli un novērst atklātās kļūdas, lai nodrošinātu stabilu un patīkamu lietotāja pieredzi.
8. Izveidot lietotāja rokasgrāmatu un apmācību materiālus, lai palīdzētu spēlētājiem ātri iepazīties ar spēles funkcionalitāti un iespējām.

Akronīmi un saīsinājumi

|  |  |
| --- | --- |
| *AI* | Mākslīgais intelekts (*Artificial Intelligence*) - Ģenerē dinamiskas kombinācijas un pārvalda spēles loģiku, lai uzlabotu spēlētāja pieredzi. |
| *API* | Lietojumprogrammu programmēšanas saskarne (*Application Programming Interface*) - Nodrošina saziņu starp spēles klienta un servera komponentēm. |
| *CSS* | Kaskadējošās stila lapas (*Cascading Style Sheets*) - Veido spēles saskarni vizuāli pievilcīgu un pielāgojamu dažādām ierīcēm. |
| *CSV* | *CSV* ir datu fails, kurā dati tiek saglabāti tabulas formātā. Katra rinda pārstāv ierakstu, un lauki rindās tiek atdalīti ar komatiem vai citām atdalīšanas rakstzīmēm (piemēram, semikolu). |
| *DB* | Datu bāze (*Database*) - Datu uzglabāšanas sistēma spēlētāju kontiem, kombinācijām un citiem elementiem. |
| *ER* | Objektu un attiecību modelis (*Entity-Relationship*) - Modelis, kas atspoguļo datubāzes struktūru un dažādu objektu savstarpējās attiecības spēlē. |
| *Git* | Versiju kontroles sistēma (*Version Control System*) - Pārvalda koda versijas un veicina sadarbību izstrādes komandā. |
| *HTML* | Hipersaistes iezīmēšanas valoda (*Hypertext Markup Language*) - Veido spēles struktūru, definējot tās satura izkārtojumu. |
| *ID* | *ID* ir unikāls identifikators, ko bieži izmanto datubāzēs un programmēšanā, lai atšķirtu vienu ierakstu no citiem. |
| *JSON* | JavaScript objektu notācija (*JavaScript Object Notation*) - Datu apmaiņai starp serveri un klientu, piemēram, spēles datu vai spēlētāja progresijas saglabāšanai. |
| *NLP* | *NLP* ir mākslīgā intelekta nozare, kas nodarbojas ar datoru spēju saprast, interpretēt un ģenerēt cilvēka dabisko valodu. Tas apvieno lingvistiku, mašīnmācīšanos un datorzinātni, lai radītu algoritmus, kas spēj apstrādāt un analizēt tekstu vai runu. |
| *PHP* | Hipersaistes priekšapstrādātājs (*Hypertext Preprocessor*) - Servera puses programmēšanas valoda, kas tiek izmantota lietotāja datu apstrādei un servera-klienta komunikācijas pārvaldībai. |
| *UI* | Lietotāja saskarne (*User Interface*) - Saskarne, caur kuru lietotāji mijiedarbojas ar spēli, ievadot informāciju vai veicot kombinācijas. |
| *VS Code* | *Visual Studio Code* - Galvenais koda redaktors spēles izstrādei, piedāvā spraudņus un automatizācijas funkcijas. |
| *WEB* | Pasaules Tīmeklis (*World Wide Web*) - Platforma informācijas piekļuvei, izmantojot tīmekļa pārlūkprogrammas; tiek izmantota tiešsaistes piekļuves un funkcionalitātes aprakstīšanai. |
| *XAMPP* | *Cross-Platform, Apache, MySQL, PHP* un *Perl* - Lokāls servera risinājums spēles izstrādei un testēšanai. |

1. Uzdevuma Nostādne

Kvalifikācijas darbs ir veltīts interaktīvas *web* lapas izstrādei, kas paredzēta lietotājiem, lai tie varētu kombinēt un veidot jaunus elementus virtuālajā vidē. Projekta mērķis ir izstrādāt platformu, kurā lietotāji varēs pieteikties, izveidot profilus un piedalīties radošā darbībā, izmantojot pieejamos rīkus un funkcionalitāti.

* 1. Līdzīgu programmatūru analīze
     1. Tiešsaistes spēle “*Little Alchemy 2*”

**Produkta apraksts**

"*Little* *Alchemy* *2*" ir populāra tiešsaistes spēle, kas piedāvā lietotājiem iespēju izveidot dažādus elementus, kombinējot četrus sākotnējos elementus – uguni, zemi, ūdeni un gaisu. Spēles galvenais mērķis ir atklāt un izveidot jaunus objektus, kas veidojas no šīm kombinācijām, sākot no vienkāršiem dabas elementiem līdz sarežģītiem konceptiem un objektiem.

Lai piekļūtu spēlei, nav nepieciešama reģistrācija (skat. *1*. attēlu), tādējādi lietotājiem ir iespēja ātri sākt spēlēt bez sarežģītiem konta izveides soļiem. Tomēr reģistrētiem lietotājiem tiek nodrošināta papildu funkcionalitāte – piemēram, progress tiek saglabāts un ir iespējams turpināt no vietas, kur lietotājs pārtrauca (skat. *2*. attēlu). Šāda sistēma mudina lietotājus atgriezties spēlē, veidojot ilgtspējīgu spēlēšanas pieredzi.

A screenshot of a video game

Description automatically generated

1. att. “*Little Alchemy 2*”

**A screenshot of a computer

Description automatically generated**

2. att. “*Little Alchemy 2*” Reģistrēšanās lapa

* + 1. Tiešsaistes spēle “*Atomas*”

**Produkta apraksts**

**Atomas** ir aizraujoša un stratēģiska puzzle spēle, kuras pamatā ir ķīmisko elementu kombinēšana. Spēles galvenais mērķis ir apvienot atomus, lai izveidotu smagākus elementus, pamatojoties uz periodisko tabulu. Ar vienkāršu, bet stratēģisku mehāniku spēle piesaista spēlētājus.

Spēles sākumā spēlētājam tiek dots aplis, kurā atrodas viegli ķīmiskie elementi, piemēram, ūdeņradis vai hēlijs (skat. 3.att). Spēlētāja uzdevums ir apvienot šos elementus, lai izveidotu smagākus, piemēram, litiju vai skābekli. Elementi tiek apvienoti, tos novietojot blakus viens otram. Kad divi vienādi elementi atrodas blakus, tos var savienot, izveidojot vienu jaunu elementu, kas ir smagāks par iepriekšējo (skat. 4.att.).

A blue circle with white text and a symbol in center

Description automatically generated with medium confidence

3. att. “*Atomas*”

A screenshot of a cell phone

Description automatically generated

4. att. “*Atomas*” Pamat elementi

* + 1. Tiešsaistes spēle “*****Doodle God*****”

**Produkta apraksts**

**Spēlē “Doodle God”** spēlētāji apvieno dažādus elementus, lai izveidotu jaunus objektus un pasaules koncepcijas. Sākot ar pamatelementiem, piemēram, ūdeni, zemi, gaisu un uguni, spēlētāji var kombinēt šos elementus, lai atklātu sarežģītākus radījumus, tehnoloģijas un pat civilizācijas. Spēlei piemīt radošuma potenciāls, un tās loģiskā struktūra pieprasa izdomāt efektīvas kombinācijas, lai atklātu jaunus objektus un paplašinātu pasauli.

Spēle arī iekļauj dažādus izaicinājumus, kas piešķir spēlei papildu dziļumu. Spēlētāji var piedalīties dažādos misiju režīmos, piemēram, izdzīvošanas vai mitoloģiskajos režīmos, kur viņiem ir jāveido noteikti objekti, lai izpildītu uzdevumus. Šī interaktivitāte ļauj spēlei saglabāties svaigai un aizraujošai ilgtermiņā (skat. 6.att.).

A screenshot of a game

Description automatically generated

5. att. “*Doodle* *God*” Izaicinājumi

Kvalifikācijas darba izstrādātajai *Web* spēlei jānodrošina šādas funkcionālās prasības:

* Lietotāju reģistrācija – iespēja lietotājiem izveidot kontu, norādot nepieciešamos datus, lai piekļūtu platformai.
* Lietotāju profilu izveide un pārvaldība – lietotāji varēs izveidot un pielāgot savus profilus, lai personalizētu savu pieredzi sistēmā.
* Elementu kombinēšana – lietotājiem jābūt iespējai virtuālajā vidē veidot jaunus objektus, apvienojot pieejamos elementus, izmantojot intuitīvu saskarsni.
* Datu glabāšana un atjaunošana – lietotājiem jābūt iespējai saglabāt un pārskatīt savus iepriekš veiktos darbus, kā arī atjaunot datus, ja tie kļūdaini mainīti.
* Navigācijas iespējas – jānodrošina viegli pieejama navigācijas sistēma, kas ļauj lietotājiem piekļūt saviem darbiem, rīkiem un citu lietotāju izveidotajiem elementiem.

1. Prasību specifikācija
   1. Ievads
      1. Dokumenta nolūks

Šis dokuments ir izstrādāts, lai aprakstītu un noteiktu prasības projektam "B.Iespējas". Tas kalpo kā pamats projekta attīstībai, sniedzot skaidru izpratni par funkcionalitāti, lietotājiem un ierobežojumiem.

* + 1. Darbības sfēra

Projekts "B.Iespējas" ir paredzēts spēlētājiem, kuri vēlas piedalīties radošā un interaktīvā vidē, kurā viņi var kombinēt pamatelementus, lai radītu jaunus priekšmetus. Tas nodrošina vispārēju spēļu pieredzi, kur nav noteikti mērķi, tā vietā veicinot radošumu.

* + 1. Definīcijas

**B.Iespējas**: Spēle, kurā spēlētāji kombinē dažādus elementus, lai radītu jaunus priekšmetus un paplašinātu savu elementu bibliotēku, balstoties uz loģiku un radošumu.

**Pamatelementi**: Uguns, ūdens, zeme un gaiss – četri sākotnējie elementi, kas pieejami katram spēlētājam, un kuri veido pamatu jaunu elementu kombinēšanai.

**AI**: Mākslīgais intelekts, kas spēlē izmantots, lai nodrošinātu inteliģentu kombināciju apstrādi, ģenerējot jaunus elementus, un nodrošinātu loģiskas un radošas kombināciju iespējas, pamatojoties uz lietotāja ievadi.

**Lietotājs**: Persona, kas reģistrējas un lieto WEB lapu, lai spēlētu spēli, veidotu jaunus elementus un saglabātu progresu personīgajā profilā.

**Saglabāto elementu bibliotēka**: Lietotāja veidoto un saglabāto elementu kolekcija, kuru iespējams izmantot atkārtoti spēles gaitā.

**Kombinēšanas loģika**: Noteikumu kopums, kas nosaka, kādi jauni elementi tiek radīti, pamatojoties uz spēlētāja izvēlētajiem diviem sākotnējiem vai jau izveidotajiem elementiem.

**Reģistrēšanās un pierakstīšanās**: Funkcionalitāte, kas nodrošina lietotāja profila izveidi un piekļuvi saglabātajiem elementiem, lai nodrošinātu personalizētu spēles pieredzi.

**Datubāze**: Sistēmas daļa, kurā tiek glabāti lietotāju profili, radītie un saglabātie elementi, nodrošinot iespēju turpināt spēli no pēdējās sesijas.

* + 1. Dokumenta pārskats

Šis dokuments apraksta prasības, kas nepieciešamas spēles prototipa izveidei. Sākumā tiek definēts dokumenta mērķis un darbības joma, kā arī sniegti skaidrojumi būtiskākajiem terminiem, kas tiks izmantoti turpmākajās sadaļās. Tālāk tiek sniegts detalizēts funkcionālo un nefunkcionālo prasību apraksts, ietverot lietotāju reģistrācijas un pieteikšanās procesu, uzdevumu piešķiršanas un pārraudzības mehānismus, kā arī lietotāja saskarnes darbību. Šī dokumenta galvenais uzdevums ir nodrošināt skaidru un vienotu izpratni starp izstrādes komandu un citām iesaistītajām pusēm, kā arī kalpot par ceļvedi sistēmas izstrādes laikā.

* 1. Vispārīgs apraksts
     1. Spēles perspektīva

Šis projekts “B.Iespējas” ir izveidots, lai piedāvātu interaktīvu un radošu vidi spēlētājiem, kuri vēlas tiešsaistē izzināt neierobežotas kombinācijas un atklāt jaunus elementus un objektus smilškastes spēļu pasaulē.

* + 1. Spēles funkcijas

**1. Reģistrācija un pierakstīšanās**

Lietotāja reģistrācija – iespēja izveidot jaunu lietotāja profilu, norādot e-pastu un paroli.

Lietotāja pierakstīšanās – reģistrēts lietotājs var pieteikties sistēmā, izmantojot savu e-pastu un paroli.

Sesijas pārvaldība – pēc veiksmīgas pierakstīšanās lietotāja sesija tiek saglabāta, nodrošinot piekļuvi personīgajiem datiem un saglabātajiem elementiem.

**2. Elementu pārvaldība**

Pamatelementu pieejamība – lietotājam vienmēr ir pieejami četri pamatelementi: uguns, ūdens, zeme un gaiss.

Elementu veidošana – lietotājs var kombinēt divus elementus, lai radītu jaunu elementu, izmantojot radošu un loģisku kombināciju sistēmu, kas tiek atbalstīta ar mākslīgā intelekta algoritmu.

Elementu saglabāšana – lietotājam ir iespēja saglabāt izveidotos elementus savā profilā turpmākai izmantošanai.

**3. Saglabāto elementu bibliotēka**

Personīgā elementu bibliotēka – lietotājam ir pieejama sadaļa, kurā var apskatīt un pārvaldīt visus saglabātos elementus.

Meklēšanas funkcija – iespēja ātri atrast konkrētu elementu, izmantojot meklēšanas joslu.

**4. Spēles funkcionalitāte**

Elementu kombinēšana – lietotājs var vilkt un nomest elementus spēles laukumā, lai tos kombinētu un radītu jaunus elementus.

Elementu vilkšana un nomešana – spēles laukums nodrošina elementu pārvietošanu un atrašanās vietas maiņu, lai veicinātu radošu eksperimentēšanu.

Elementu dzēšana – lietotājs var izdzēst elementus, tos velkot uz “Notīrīt” zonu.

Elementu saglabāšana spēles laikā – lietotājs var saglabāt izveidoto elementu, velkot to uz “Saglabāt” zonu.

**5. Uzstādījumi**

Sākt jaunu spēli – lietotājs var sākt jaunu spēli, kurā tiek izdzēsti visi esošie elementi no spēles laukuma.

Turpināt spēli – lietotājs var turpināt iepriekšējo spēli, izmantojot iepriekš saglabātos elementus.

Izrakstīšanās – lietotājam ir iespēja iziet no profila, iznīcinot aktīvo sesiju.

**6. Datubāzes pārvaldība**

Lietotāju datu glabāšana – datubāzē tiek glabāti visi reģistrētie lietotāji ar drošu paroļu šifrēšanu.

Elementu glabāšana – datubāze nodrošina izveidoto un saglabāto elementu uzskaiti un pārvaldību katram lietotājam individuāli.

* + 1. Lietotāja raksturiezīmes

Sistēma "B.Iespējas" ir paredzēta dažādiem lietotājiem, neatkarīgi no viņu tehniskajām zināšanām vai pieredzes. Lietotājiem tiek nodrošināta draudzīga un intuitīva lietotāja saskarne, kas ļauj ātri apgūt sistēmas funkcionalitāti.

* + 1. Spēles līmeņi

Produktam ir paredzēta tikai viena līmeņa piekļuve web daļā, un tā ir lietotājs. Administrēšanas funkcijas tiks veiktas tieši datubāzes līmenī, un atsevišķa vizuālā web daļa administratoram netiks izstrādāta, jo administrēšanai nepieciešamās darbības nav paredzēts veikt no web puses.

* **Līmenis – reģistrētais lietotājs**
  + Reģistrētajiem lietotājiem būs iespēja:
  + Izveidot un pārvaldīt savu profilu.
  + Kombinēt pamata elementus un veidot jaunus, unikālus elementus.
  + Saglabāt izveidotos elementus un pārskatīt savu personalizēto elementu bibliotēku.
  + Veidot vairākas spēļu sesijas, izmantojot dažādus saglabātos elementus.
  + Turpināt iepriekšējās spēļu sesijas, saglabājot progresu datubāzē.
* Līmenis – administrators

Administratoram ir piekļuve datubāzei un tajā glabātajiem datiem, kā arī iespēja rediģēt un veikt ar tiem nepieciešamās darbības konkrētās situācijās. Administratora atbildība ir nodrošināt datu drošību un integritāti, kā arī pārvaldīt lietotāju kontus un saglabātos elementus.

* + 1. darbības apraksts

**Reģistrācija** **un** **pierakstīšanās**: Lietotājs izveido profilu, ievadot e-pastu un paroli, un pēc tam var pierakstīties, lai piekļūtu saviem saglabātajiem elementiem.

**Elementu kombinēšana**: Spēlētājs kombinē pamatelementus spēles laukumā, radot jaunus elementus, kurus ģenerē AI algoritms.

**Elementu saglabāšana un dzēšana**: Jauni elementi var tikt saglabāti, velkot tos uz saglabāšanas zonu, vai dzēsti, velkot uz notīrīšanas zonu.

**Spēļu sesiju saglabāšana**: Lietotājs var turpināt iepriekš uzsākto spēli.

**Bibliotēka**: Saglabātie elementi tiek glabāti datubāzē un pieejami bibliotēkā ar meklēšanas funkciju.

**Administrēšana**: Administrators pārvalda datubāzi un veic nepieciešamās izmaiņas datu drošībai un integritātei.

* + 1. Ierobežojumi
* Interneta pieslēgums: Spēlei nepieciešama aktīva interneta pieslēguma piekļuve.
* Tehniskie ierobežojumi: Web lietotnei ir jānodrošina atbilstoša optimizācija, lai tā vienlīdz labi darbotos dažādās ierīcēs, pārlūkos un operētājsistēmās. Nepieciešama arī stabila veiktspēja, ja vienlaicīgi pieslēdzas vairāki lietotāji.
* Valodas pieejamība: Spēle ir pieejama Latviešu valodā.
  + 1. Klienta prasības
* Funkcionālās prasības

**Reģistrācijas un pierakstīšanās funkcionalitāte**: Lietotājam jābūt iespējai reģistrēties un pierakstīties, izmantojot e-pastu un paroli.

**Elementu kombinēšana**: Lietotājs var kombinēt pamatelementus, lai radītu jaunus elementus, un izmantot tos turpmākai kombinēšanai.

**Elementu saglabāšana**: Lietotājam jābūt iespējai saglabāt izveidotos elementus, lai tos izmantotu vēlāk.

**Personīgā bibliotēka**: Saglabātie elementi jāparāda lietotājam bibliotēkā ar iespēju tos meklēt un pārskatīt.

**Spēļu sesiju pārvaldība**: Lietotājam jābūt iespējai sākt jaunu spēli vai turpināt iepriekšējo, izmantojot saglabātos elementus.

**Notīrīšanas funkcija**: Spēles laukam jābūt iespējai dzēst elementus, velkot tos uz speciālu notīrīšanas zonu.

**Saglabāšanas funkcija**: Jānodrošina vienkāršs mehānisms elementu saglabāšanai spēles laikā, velkot elementu uz saglabāšanas zonu.

* Nefunkcionālās prasības

**Veiktspēja**: Sistēmai jānodrošina stabila darbība, lai tā ātri reaģētu uz lietotāja darbībām un kombinētu elementus reālajā laikā.

**Lietojamība**: Spēlei jābūt vienkāršai un intuitīvai, lai to varētu izmantot dažādu vecumu un prasmju lietotāji.

**Drošība**: Lietotāju dati, tostarp e-pasti un paroles, jāaizsargā, izmantojot drošas šifrēšanas metodes.

* Dizaina prasības

**Pievilcīgs un vienkāršs dizains**: Lietotāja interfeisam jābūt tīram, pievilcīgam un viegli saprotamam.

**Radošs vizuālais stils**: Elementiem un to kombināciju rezultātiem jābūt attēlotiem vizuāli pievilcīgā un saprotamā formā.

**Funkciju izcelšana**: Svarīgās funkcijas, piemēram, saglabāšana un notīrīšana, jāizceļ, lai tās būtu viegli pamanāmas.

* 1. Funkcionālais un dinamiskais sistēmas modelis
     1. Lietojumgadījumu diagramma

Lietojumgadījumu diagramma izmantota, lai vizualizētu spēles sistēmas funkcionalitātes, kas pieejamas dažādiem lietotājiem. Šī diagramma palīdz analizēt un plānot spēles attīstības procesu.

Diagrammā attēloti 2 galvenie lietotāji, kas mijiedarbojas ar sistēmu (skat. 6. att.):

* Reģistrēts spēlētājs - lietotājs, kurš ir reģistrējies spēlē un piekļūst visām tās funkcijām.
* Jauns spēlētājs - persona, kura vēl nav reģistrējusies.

Administrators ir tehniskais lietotājs, kurš nav attēlots diagrammā, jo tā darbības notiek datubāzes līmenī, piemēram, spēlētāju datu pārvaldība, sūdzību apstrāde un resursu balansa pielāgošana.

A diagram of a structure

Description automatically generated

6. att. Lietojumgadījumu diagramma

Spēlētāji mijiedarbojas ar lietojumgadījumiem (use-cases), kas diagrammā attēloti kā ovālas formas simboli. Mijiedarbošanās veidi norādīti ar trim dažādiem līnijām:

* Nepārtrauktās līnijas - apzīmē obligātās darbības, piemēram, reģistrācija vai spēlētāja konta izveidošana, lai piekļūtu pilnīgai spēles pieredzei.
* Raustītās līnijas - norāda uz izvēles darbībām, piemēram elementa saglabāšana. Šīs darbības nav obligātas, bet tās paplašina spēles iespējas.
* Līnijas ar apli galā - apzīmē pārejas starp lietojumgadījumiem ar konkrētiem nosacījumiem..

Papildu darbību un pāreju skaidrojumi atrodas ovālas formas simbolos blakus attiecīgajām līnijām diagrammā. Tie sniedz detalizētu informāciju par spēles loģiku un nosacījumiem, kas nepieciešami dažādu funkcionalitāšu aktivizēšanai.

* 1. Funkcionālās prasības
     1. Lietotāja reģistrācija

Lietotāja reģistrācija ir nepieciešama, lai lietotājs varētu pilnvērtīgi izmantot lapas sniegtās iespējas.

**Izsaukšanas elements:** Poga "Reģistrēties"

**Ieejas parametri:** e-pasts, parole, atkārtota parole

Darbība (skat. 6. att.)

A diagram with black text

Description automatically generated

7. att. Reģistrēšanās funkcijas blokshēma

**Pieteikšanās:** Lietotājs noklikšķina uz pogas "Reģistrēties"

**Informācijas nolasīšana:** Nolasa ievadītos datus e-pasts, parole, atkārtota parole, pārbauda, vai visi lauki ir aizpildīti.Ja kāds no laukiem nav aizpildīts, parādās paziņojums: "Visiem laukumiem jābūt aizpildītiem!"

**Datu pārbaude:** Pārbauda ievadīto datu pareizību un atbilstību noteiktajiem kritērijiem:

* Pārbauda, vai e-pasts jau nav datubāzē.
* Pārbauda, vai abas ievadītās paroles sakrīt.

Ja e-pasts jau eksistē datubāzē, parādās paziņojums: "Šis e-pasts jau tiek izmantots!" Ja ievadītās paroles nesakrīt, parādās paziņojums: "Paroles nesakrīt!"

**Konta izveide:** Kad dati ir veiksmīgi pārbaudīti un apstiprināti, tiek izveidots konts, un lietotājs tiek pievienots datu bāzei ar unikālu identifikatoru.

**Paziņojums par veiksmīgu reģistrāciju:** Lietotājam tiek parādīts paziņojums, kas norāda, ka reģistrācija ir veiksmīga.

**Loga aktivizēšana:** Pēc veiksmīgas reģistrācijas tiek aktivizēts "Sākumlapas logs", ļaujot lietotājam sākt izmantot lapas funkcijas.

* + 1. Lietotāja pieteikšanās

Lietotāja pieteikšanās ir nepieciešama, lai lietotājs varētu pilnvērtīgi piekļūt sistēmas funkcijām.

**Izsaukšanas elements:** Poga "Pieteikties"

**Ieejas parametri:** e-pasts un parole

Darbība (skat. 7. att.)

A diagram of a flowchart

Description automatically generated

8. att. Pieteikšanās funkcijas blokshēma

**Pieteikšanās:** Lietotājs noklikšķina uz pogas "Pieteikties".

**Informācijas nolasīšana:** Nolasa ievadītos datus - e-pastu un paroli, pārbauda, vai abi lauki ir aizpildīti. Ja kāds no laukiem nav aizpildīts, parādās paziņojums: "Visiem laukumiem jābūt aizpildītiem!"

**Datu pārbaude:** Pārbauda ievadīto datu pareizību un atbilstību noteiktajiem kritērijiem:

* Pārbauda, vai e-pasts eksistē datubāzē.
* Pārbauda, vai parole ir pareiza un atbilst e-pastam.

Ja e-pasts vai parole nesakrīt ar datubāzes datiem, parādās paziņojums: "E-pasts vai parole ir nepareizi!"

**Autentifikācija:** Ja ievadītie dati sakrīt ar datubāzē esošajiem, lietotājs tiek autentificēts un pieteikts sistēmā.

**Loga aktivizēšana:** Pēc veiksmīgas pieteikšanās tiek aktivizēts "Sākumlapas logs", ļaujot lietotājam piekļūt saviem datiem un citām sistēmas funkcijām.

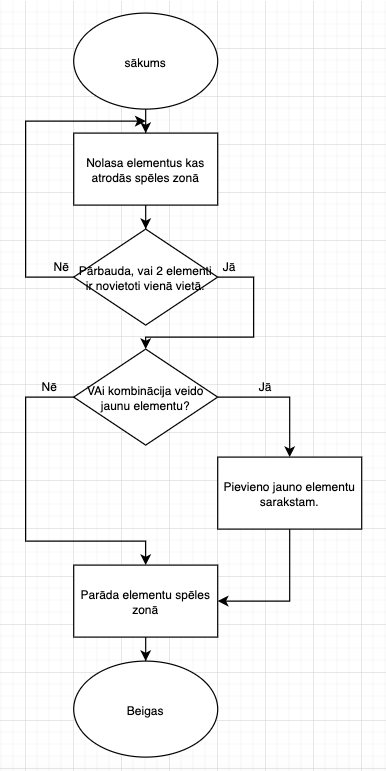
* + 1. Spēles elementu kombinācijas un rezultātu ģenerēšana

Elementu kombinācija un rezultātu ģenerēšana ļauj spēlētājiem apvienot dažādus elementus, lai radītu jaunus objektus vai materiālus.

**Izsaukšanas elements**: Elementu vilkšana no izvēlnes uz darba zonu.

**Ieejas parametri**: Divi elementi.

Darbība (skat. 8. att.)



9. att. kombinācijas funkcijas blokshēma

**Elementu vilkšana:** Spēlētājs izvēlas divus elementus no izvēlnes un vilks tos uz darba zonu.

**Elementu novietošana:** Kad elementi tiek novietoti darba zonā vienā vietā, tie tiek apstrādāti, lai pārbaudītu, vai kombinācija jau eksistē.

**Kombinācijas pārbaude:** Spēles sistēma pārbauda, vai kombinētie elementi rada jaunu, spēlē pieejamu elementu. Ja kombinācija ir pareiza, tiek ģenerēts jauns elements.

**Jauna elementa radīšana:** Ja kombinācija ir jauna, tiek parādīts jauns elements, kas tiek pievienots spēlētāja elementu sarakstam.

**Eksistējoša elementa radīšana**: Ja kombinācija jau eksistē, tiek parādīts eksistējošais elements, kas netiek pievienots spēlētāja elementu sarakstam atkārtoti.

* + 1. Elementu saglabāšana

Elementu saglabāšana ļauj spēlētājiem saglabāt jaunus elementus biblotekā.

**Izsaukšanas elements:** Poga “Saglabāt”.

**Ieejas parametri:** Jauni elementi, kas iegūti no kombinācijām.

Darbība (skat. 9. att.)

A diagram of a diagram

Description automatically generated

10. att. Jaunu elementu funkcijas blokshēma

**Elementa pievienošana:** Lietotājs noklikšķina pogu “Saglabāt”.

**Datu sagatavošana:** Pārbauda vai elements jau saglabāts.

**Pievienošana:** Ja elements vēl nav saglabāts tas tiek pievienots biblotekai.

* + 1. Elementu meklēšanas funkcija

Elementu meklēšanas funkcija ļauj lietotājiem ātri atrast konkrētus elementus no spēles elementu saraksta.

**Izsaukšanas elements:** Meklēšanas josla.

**Ieejas parametri:** Teksts, kas ievadīts meklēšanas joslā.

Darbība (skat. 10. att.)

A diagram of a data flow

Description automatically generated

11. att. kombinācijas funkcijas blokshēma

**Meklēšanas josla:** Spēlētājs ievada meklējamo elementa nosaukumu vai daļu no nosaukuma meklēšanas joslā.

**Rezultātu filtrēšana:** Spēles sistēma automātiski filtrē elementus atbilstoši ievadītajam tekstam un parāda atbilstošos rezultātus.

**Meklēšanas rezultāti:** Meklēšanas rezultāti tiek parādīti kā saraksts ar elementiem, kas atbilst ievadītajiem kritērijiem.

* 1. Nefunkcionālās prasības
     1. Kopējās nefunkcionālās prasības
        1. Standarta poga

Pogas (skat. 11. att.) domātas funkciju veikšanai, tās pārsvarā izskatās vienādi izņemtot dažos speciālos gadijumos, piemēram kā atpakaļ poga.



12. att. Pogas pirms pārbraukšanas ar kursoru

* Poga ir taisnstūrveida
* Fona krāsa: zila (#007bff);
* Teksta krāsa: balta (#FFFFFF);
* Teksta izmērs: 18 px;
* Teksta novietojums: pa vidu pogai, centrēts;

Pārbraucot pār pogu ar kursoru tā mainās: (skat.12. att.):



13. att. Poga pēc pārbraukšanas ar kursoru

* Fona krāsa: tumši zila (#0056b3);
* Izņēmums – Atgriesties poga Fona krāsa: sarkana (#ff5050)
  + - 1. Satura virsraksti

Katrā logā atrodas saturiskais virsraksts (skat.13. att.), lai lietotājiem būtu iespējams saprast, kas atvērtajā lapā tiks parādīts. Visu programmas virsrakstu izskats ir līdzīgs:

* Teksta krāsa: balts (#FFFFFF)*;*
* Teksta izmērs: 16 px;
* Teksta fonts:Verdana;
* Teksta efekts: treknraksts;



14. att. Loga satura virsraksta makets

* + - 1. Ievadlauki

Ievadlauki paredzēti informācijas ievadīšanai un to izskats ir līdzīgs (skat. 11. att.).



15. att. Ievadlauks pirms tajā ieklikšķināts

* Ievadlauku augstums: 2 rem;
* Ievadlauku platums: 4 rem;
* Ievadlauku stūri: Noapaļoti;
* Fona krāsa: balta rgba(255, 255, 255, 0.2) ;
* Teksta krāsa: balta (*#000000);*
* Teksta fonts:Verdana;
* Teksta izmērs: 1 rem;
* Teksta novietojums: ievadlauka kreisajā malā
* Rāmja biezums: 4 px;
* Rāmja krāsa, ja ievadlaukā ieklikšķina ar datorpeli: tumši zila (*#244DB3)*
* Ja ievadlauks paredzēts parolēm: informāciju slēpj, rādot “\*” simbolu (skat. 12. att.);



16. att. Ievadlauka, ja tajā raksta paroli

* + - 1. Paroles parametra izskats ievades laikā
* Teksta krāsa: zaļa (#00ff7f)
* Fonta izmērs: 0.9 rem
* Novietojums: tieši zem paroles lauka

17. att. Ja prasības izspildīts:

* Teksta krāsa: sarkana (#ff4d4d)
* Fonta izmērs: 0.9 rem
* Novietojums: tieši zem paroles lauka

18. att. Ja prasības nav izpildītas:

* + - 1. Ievadlauku nosaukumi

Pirms katra ievadlauka atrodas teksts (skat. 13. att.), kurš paskaidro, kādai informācijai ievades lauks paredzēts.

* Teksta krāsa: balta (#FFFFFF);
* Teksta izmērs: 16 px;
* Teksta fonts:Verdana;
* Apakšējā: 0,5 rem no nākamā elementa



19. att. Ievadlauka nosaukums

* + - 1. Elements

Spēlējot var izveidot bezgalīgi daudz elementus (skat. 17. att.), kā arī saglabāt savus mīļākos. Tie parāda kāds elements tika izveidot kopā ar emoji kurš viss labāk attēlo šo elementu.

* Poga ir taisnstūrveida;
* Pogas izmērs piemērojās teksta garumam;
* Fona krāsa: rgba(255, 255, 255, 0.8);
* Teksta fonts:Verdana;
* Teksta krāsa: balta (#FFFFFF);
* Teksta izmērs: 1 rem;
* Teksta novietojums: pa vidu pogai, centrēts;
* Emoji novietojums: pa kreisi no teksta



20. att. Elements

* + 1. Reģistrācijas logs

Reģistrācijas logs (skat. 18. att.) ir lietotāja saskarnes (interfeisa) komponents, kas tiek izmantots, lai jauni lietotāji varētu izveidot savu kontus.

A screenshot of a computer

Description automatically generated

21. att. Reģistrācijas logs

Lapas vidū ir satura virsraksts ar tekstu “Reģistrēties” (Satura virsrakstu apskatīti 2.4.1.3. punktā) zem tā redzemi 3 ievades lauki (ievadlauki apskatīti 2.4.1.3. punktā) virs katra ievadlauka ir nosaukums kurš norāda kāda informācija tiek prasīta (ievadlauka nosaukumus apskatīti 2.4.1.4. punktā).

Ievadlauki:

* E-pasts;
* Parole;
* Atkārtot paroli.

Zem pēdējā ievadlauka “Atkārtot paroli” atrodas poga “Reģistrēties” (pogas apskatīt 2.4.1.2)

Zem pogas atrodas teksts “Jau esi reģistrējies?”, kur pasvītrotā daļa ir hipersaite, kas lietotāju aizved uz pieteikšanās logu.

* Teksta izmērs: 1.2 rem;
* Teksta krāsa: balta (#FFFFFF);
* Atstarpes starp ievadlaukiem: 0,5 rem;
  + 1. Pieteikšanās logs

Pieteikšanās logs (skat. 14. att.) ir domāts lietotājiem kuriem jau ir aktīvs profils. Izskata ziņā viņš ir identisks Reģistrācijas logam izņemot:

* Satura virsraksta teksts ir “Pieteikšanās”
* Nav “Atkārtot paroli” ievadlauks;
* Zem pēdējā lauka ir poga “Pieteikties”;
* Zem pogas atrodās teksts “Nav izveidots profils?”, kur pasvītrotā daļa ir hipersaite, kas lietotāju aizved uz reģistrācijas logu.

A screenshot of a computer

Description automatically generated

22. att. Pieteikšanās logs

* + 1. Sākumlapas logs

Sākumlapā (skat.19. att.) ir redzams satura virsraksts ar tekstu “B.Iespējas” (Satura virsrakstu apskatīti 2.4.1.3. punktā). Zem virsraksta ir dotas 4 pogas:

1. “Turpināt”;
2. “Jauna spēle”;
3. “Bibloteka”;

Pogu izskats:

* Pogas ir taisnstūrveida
* Fona krāsa: zila (#007bff);
* Teksta krāsa: balta (#FFFFFF);
* Teksta izmērs: 20 px;
* Teksta novietojums: pa vidu pogai, centrēts;

Pārbraucot pār pogu ar kursoru tā mainās:

* Fona krāsa: tumši zila (#0056b3);
* “Log out” (pogas apskatīt 2.4.1.2).

A screenshot of a computer

Description automatically generated

23. att. Sākumlapas logs

* + 1. Spēles logs

Spēles logs (skat. 20. att.) ir balta zona kura uz kuras ir redzamas 3 pogas un labajā pusē ir elementu saraksts.

Spēles zonas pogas (pogas apskatīt 2.4.1.2):

1. “Atgriesties” Atrodās: augšējā kreisajā stūrī;
2. “Notīrīt” Atrodās: apakšējā kreisajā stūrī;
3. “Saglabāt” Atrodās: apakšējā labajā stūrī, pa kreisi no elementa saraksta.

Elementa saraksta izskats:

* Augšējā: 0 px;
* Apakšējā: 0 px;
* No labās puses: 0 px;
* Augstums: 100% no lapas loga;
* Platums: Regulējams; standarta – 15 rem no lapas loga.

Elementu sarakstā atrodās meklēšanas josla un šķirošanas poga (pogas apskatīt 2.4.1.2) zem tiem ir redzami elementi (elementu apskatīt 2.4.1.6) kurus ir iespēja izvilkt uz spēles zonas.

A screenshot of a computer

Description automatically generated

24. att. Spēles logs

* + 1. Biblotekas logs

Biblotekā (skat. 21. att.) ir redzama “Atgriesties” poga (pogas apskatīt 2.4.1.2), tā atrodās: augšējā kreisajā stūrī.

Ekrāna vidū ir redzams satura virsraksts ar tekstu “Bibloteka” (Satura virsrakstu apskatīti 2.4.1.3. punktā).

Zem virsraksta ir kastīte

kastītes parametri:

* Platums: 30 rem;
* Fona krāsa: rgba(255, 255, 255, 0.2);
* Teksta krāsa: balta ( #FFFFFF);
* Teksta fonts:Verdana;
* Rāmja krāsa: balta (#FFFFFF);
* Rāmja biezums: 2px;
* 1 rem zem attēla atrodas meklēšanas josla;
* 2 rem zem meklēšanas joslas atrodas elementi (elementu apskatīt 2.4.1.6).

A screenshot of a computer

Description automatically generated

25. att. Biblotekas logs

1. uzdevumu risināšanas līdzekļu pamantojums

Spēļu izstrādes vides iekārtošana **B.Iespējas** radīšanai.

Lai radītu **B.Iespējas**, tika izveidota daudzpusīga izstrādes vide, kas atbalsta gan kreatīvo procesu, gan tehnisko realizāciju. Šeit ir detalizēts pārskats par izmantotajiem rīkiem un tehnoloģijām.

* 1. Koda redaktors
* ***Visual Studio Code***

***Visual Studio Code* ir viegla un jaudīga koda rediģēšanas platforma, ko izstrādājusi Microsoft, nodrošinot programmētājiem plašas iespējas, tostarp vairāku programmēšanas valodu atbalstu un paplašināšanas iespējas ar spraudņiem.**

**Projekta B.Iespējas izstrādes laikā tika izmantots Visual Studio Code, jo tas piedāvā lietotājam draudzīgu saskarsmi, kas atvieglo darbu ar dažādu programmēšanas valodu failiem, piemēram, HTML, CSS un JavaScript. Redaktors nodrošina integrētu termināli un atkļūdošanas iespējas, kas ir būtiski koda testēšanai un labošanai. Tā saderība ar dažādām operētājsistēmām, tostarp Windows un macOS, padara to par ideālu rīku šim projektam.**

* 1. Mākslīgais intelekts
* ***OpenAI Platform***

***OpenAI* platforma ir uz mākoņpakalpojumiem balstīts mākslīgā intelekta rīks, kas nodrošina piekļuvi dažādiem valodas modeļiem, piemēram, *GPT-3* un *GPT-4*. Tā piedāvā *API*, kas ļauj izstrādātājiem integrēt dabiskās valodas apstrādes (NLP) funkcionalitāti savās lietotnēs un sistēmās.**

**B.Iespējas projekta ietvaros *OpenAI* platforma tika izvēlēta, jo tā piedāvā ērtu veidu, kā ģenerēt un apstrādāt tekstu, balstoties uz lietotāja ievadītiem datiem. Šī platforma ļauj izveidot dinamisku spēles elementu kombinēšanas sistēmu, kas padara spēli interaktīvāku un aizraujošāku. Turklāt tās plašais pielietojuma diapazons un augstā precizitāte nodrošina, ka ģenerētie rezultāti ir loģiski un radoši, atbilstot projekta prasībām.**

* 1. Datubāzes projektēšanas rīks
* ***HeidiSQL***

***HeidiSQL* ir bezmaksas datubāzu pārvaldības rīks, kas atvieglo darbu ar datubāzēm, piemēram, *MySQL* un *MariaDB*. Tā intuitīvais dizains ļauj ātri un efektīvi izveidot un uzturēt datubāžu struktūras.**

**B.Iespējas projekta ietvaros *HeidiSQL* tika izvēlēts, jo tas piedāvā ērtu datu importēšanu, eksportēšanu un vaicājumu izpildi. Šis rīks atvieglo datubāzes tabulu apskati un labošanas procesu, kas ir būtiski, lai pielāgotos projekta prasībām. Turklāt tas nodrošina datu dublējumus un pārsūtīšanas iespējas starp sistēmām, garantējot datu drošību.**

* 1. Lokālā servera izveides rīks
* **XAMPP**

***XAMPP* ir atvērtā koda tīmekļa izstrādes komplekts, kas ietver *Apache* serveri, *MySQL* datubāzi, *PHP* un citus noderīgus rīkus. Tas ir vienkārši uzstādāms un konfigurējams, padarot to par populāru izvēli tīmekļa izstrādē.**

**Projekta B.Iespējas ietvaros *XAMPP* tika izmantots lokāla servera izveidei, kas ir nepieciešams, lai testētu *PHP* un *MySQL* bāzētas sistēmas. Šis rīks nodrošina stabilu vidi, kurā simulēt servera darbību un pārbaudīt spēles funkcionalitāti.**

* 1. Blokshēmu izveides rīks
* ***Mind on map***

***Mind on map* ir tīmekļa rīks, kas atvieglo diagrammu un blokshēmu veidošanu. Tā plašās veidņu un pielāgošanas iespējas ļauj precīzi vizualizēt projekta loģiku un datu plūsmas.**

**Projekta B.Iespējas ietvaros tas tika izmantots, lai izveidotu strukturētas blokshēmas, kas skaidri attēlo spēles sistēmu un procesus, nodrošinot efektīvu izstrādātāju un dizaineru sadarbību.**

* 1. Programmēšanas valodas
* ***HTML* (*HyperText Markup Language*)**

***HTML* ir pamata marķēšanas valoda, kas nosaka tīmekļa lapu satura struktūru.**

**Projekta B.Iespējas izstrādes laikā *HTML* tika izmantots spēles tīmekļa komponentu veidošanai, nodrošinot stabilu pamatu satura strukturēšanai.**

* ***JavaScript***

**JavaScript ir skriptu valoda, kas tiek izmantota interaktivitātes un dinamiskas funkcionalitātes nodrošināšanai.**

**Spēles B.Iespējas izstrādē *JavaScript* tika izmantots, lai izveidotu dinamiskas funkcijas, piemēram, reāllaika mijiedarbību un spēles elementu animācijas.**

* ***CSS* (*Cascading Style Sheets*)**

***CSS* tiek izmantots, lai kontrolētu spēles tīmekļa komponentu vizuālo izskatu un dizainu.**

**Projekta ietvaros *CSS* tika izmantots, lai radītu vizuāli pievilcīgu un responsīvu dizainu, kas piemērots dažādām ierīcēm un ekrānu izmēriem.**

* ***PHP* (*Hypertext Preprocessor*)**

**PHP ir servera puses skriptu valoda, kas nodrošina dinamisku datu apstrādi un spēles servera loģiku.**

* 1. *Wireframe* izveides rīks
* ***Canva***

**Canva ir grafiskā dizaina platforma, kas piedāvā rīkus lietotāja saskarnes maketu izveidei.**

**Projekta *B.Iespējas* laikā *Canva* tika izmantota, lai radītu sākotnējos lietotāja saskarnes dizainus, kas palīdzēja skaidri definēt spēles izkārtojumu un funkcionalitāti. *Canva* nodrošina ātru un vienkāršu pielāgojumu, kas sekmē efektīvu komandas komunikāciju.**

1. Datu struktūras apraksts

Lai izstrādātu un testētu "B.Iespējas" tīmekļa aplikāciju, tiks izmantotas šādas tehnoloģijas un rīki:

* 1. Programmēšanas valodas:
* ***HTML***: Izmantots, lai veidotu lietotāja interfeisu un strukturētu aplikācijas satura izkārtojumu.
* ***CSS***: Lietots, lai stilizētu aplikāciju un nodrošinātu adaptīvu dizainu dažādām ierīcēm (*responsive design*).
* ***JavaScript***: Dinamiskai interaktivitātei, piemēram, spēles elementu vilkšanai, kombinēšanai un datu pārvaldībai spēles procesā.
* ***PHP***: Servera puses valoda, kas izmantota, lai apstrādātu lietotāju pieprasījumus, veidotu savienojumu ar datu bāzi un pārvaldītu lietotāju datus.
  1. Datu bāzes pārvaldības sistēmas:
* ***MySQL***: Izmantota, lai pārvaldītu un glabātu lietotāju kontu datus, saglabātos elementus un kombināciju vēsturi, nodrošinot efektīvu datu organizāciju un pieejamību.
  1. Izstrādes vide:
* ***Visual Studio Code***: Viegls, daudzfunkcionāls koda redaktors ar spraudņiem *HTML*, *CSS*, *JavaScript* un *PHP* izstrādei.
* ***XAMPP***: Vietējās servera vides iestatīšanai, kas ietver *Apache* serveri un *MySQL* datu bāzi, lai simulētu tīmekļa servera darbību.
  1. Papildrīki:
* ***Font Awesome*:** Ikonu bibliotēka, lai pievienotu vizuālus elementus, piemēram, ikonas spēles elementiem un interfeisa pogām.
* ***OpenAI*:** Mākslīgā intelekta elementu ģenerēšana.
  1. Datu plūsmas diagrammas

Datu plūsmas diagrammas ilustrē, kā informācija pārvietojas starp dažādiem B.Iespējas spēles komponentiem un lietotāju. Spēlei būtiskākās datu plūsmas ir šādas:

**Lietotāja autentifikācija:**

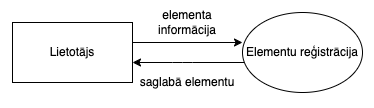
Process sākas, kad spēlētājs ievada savu lietotājvārdu un paroli. Šie dati tiek nosūtīti uz serveri, kur tie tiek pārbaudīti pret datubāzes ierakstiem. Ja dati sakrīt ar kādu no datubāzes ierakstiem, lietotājs tiek autentificēts un iegūst piekļuvi spēles funkcionalitātēm. Pretējā gadījumā spēle parāda kļūdas ziņojumu.



26. att. Lietotāja autentifikācijas datu plūsma

**Elementu reģistrācija:**

Spēlētājs ievada informāciju par savāktajiem elementiem. Šie dati tiek nosūtīti uz datubāzi un saglabāti.



27. att. Elementu reģistrācijas datu plūsma

**Meklēšana un pārvaldība:**

Spēlētājs ievada elementa nosaukumu. Sistēma veic meklēšanu datubāzē un atgriež rezultātus. Ja resursi netiek atrasti, spēlētājam tiek parādīts ziņojums par meklējamo objektu trūkumu.



28. att. Meklēšanas un pārvaldības datu plūsma

* 1. Stāvokļa diagramma

Stāvokļa diagramma parāda, kā *B.Iespējas* reaģē uz spēlētāja darbībām un maina sistēmas stāvokļus. Būtiskākie stāvokļi ir:

* **Nepieteicies spēlētājs:**
  + Pieejams tikai pieteikšanās logs.
  + Stāvokļa maiņa notiek pēc veiksmīgas autentifikācijas.
* **Autentificēts spēlētājs:**
  + Lietotājs var veidot elementus, meklēt un apskatīt tos.
* **Administratora režīms:**
  + Administrators var pārvaldīt lietotājus, skatīt datubāzes datus un veikt sistēmas konfigurācijas izmaiņas.
  1. ER modelis

Visas datubāzes tabulas ir savā starpā saistītas ar relācijām (skat. 26. att.).

Entitāšu diagramma [Autora veidota ekrānkopija]

Datubāzes "b.iespejas" galvenās tabulas ir “users”, “elements” un “saved\_elements”. Tabula “users” satur sistēmas lietotāju datus, kas saistīti ar pārējām tabulām caur users\_id lauku. Tabula elements ietver lietotāju radītus vai izmantotus elementus, bet tabula “saved\_elements” saglabā lietotāja izvēlētos elementus.

Datubāzes struktūra ir veidota ar utf8mb4 rakstzīmju kopu, kas nodrošina plašu simbolu un rakstzīmju atbalstu.

A screenshot of a computer

Description automatically generated

29. att. Entitāšu diagramma [Autora veidota ekrānkopija]

* 1. Sistēmas struktūras modelis

**Sistēmas struktūras modelis** ir būtisks projekta elements, kas palīdz vizualizēt spēles pasaules arhitektūru un tās komponentu mijiedarbību.

* + 1. Datubāzes apraksts

Datubāzes nosaukums ir “biespejas\_db”, un tā ietver 3 tabulas: **“*elements*”** (struktūru skat. 1. Tabula “*elements*”), **“*saved\_elements*”** (struktūru skat. 2. Tabula “*saved\_elements*”), **“*users*”** (struktūru skat. 3. Tabula “*users*”)

* + 1. **“*blueprints*”** tabula

Tabula "*elements*" glabā datus par visiem spēlē pieejamajiem elementiem, kurus lietotāji var radīt, kombinējot dažādus pamatelementus. Katram elementam ir unikāls identifikators (*id*) un nosaukums (*name*). Tabulā tiek reģistrēts arī elementa izveides laiks un atsauce uz lietotāju (*users\_id*), kurš elementu izveidojis.

1. Tabula “*elements*”

| **Nosaukums** | **Tips** | **Atribūti** | **Komentāri** |
| --- | --- | --- | --- |
| id | int(11) | NOT NULL, PRIMARY KEY, AUTO\_INCREMENT | Elementa unikālais identifikators |
| name | varchar(50) | NOT NULL | Elementa nosaukums |
| created\_at | timestamp | NOT NULL, DEFAULT current\_timestamp() | Elementa izveides datums |
| users\_id | int(11) | DEFAULT NULL, FOREIGN KEY (users.id) | Atsauce uz lietotāja ID |

* + 1. **“*saved\_elements*”** tabula

**Tabula "saved\_elements" glabā informāciju par lietotāju saglabātajiem elementiem. Tā nodrošina iespēju lietotājiem saglabāt un pārskatīt iepriekš izveidotos elementus. Katrs ieraksts satur unikālu identifikatoru (*id*), elementa nosaukumu (*name*), saglabāšanas datumu (*saved\_at*) un atsauci uz lietotāja ID (*users\_id*), kurš saglabājis elementu.**

2. Tabula “*saved\_elements*”

|  |  |  |  |
| --- | --- | --- | --- |
| **Nosaukums** | **Tips** | **Atribūti** | **Komentāri** |
| id | int(11) | NOT NULL, PRIMARY KEY, AUTO\_INCREMENT | Saglabātā elementa unikālais identifikators |
| name | varchar(50) | NOT NULL | Elementa nosaukums |
| saved\_at | timestamp | NOT NULL, DEFAULT current\_timestamp() | Elementa saglabāšanas datums |
| users\_id | int(11) | DEFAULT NULL, FOREIGN KEY (users.id) | Atsauce uz lietotāja ID |

* + 1. **“*users*”** tabula

Tabula "users" satur informāciju par visiem reģistrētajiem sistēmas lietotājiem. Katram lietotājam tiek piešķirts unikāls identifikators (*id*), un tiek glabāti dati par lietotāja e-pastu un paroli. Šī tabula nodrošina iespēju identificēt un autentificēt lietotājus sistēmā.

3. Tabula “*users*”

|  |  |  |  |
| --- | --- | --- | --- |
| **Nosaukums** | **Tips** | **Atribūti** | **Komentāri** |
| id | int(11) | NOT NULL, PRIMARY KEY, AUTO\_INCREMENT | Lietotāja unikālais identifikators |
| email | varchar(255) | NOT NULL, UNIQUE | Lietotāja e-pasta adrese |
| password | varchar(255) | NOT NULL | Lietotāja parole |
| created\_at | timestamp | NOT NULL, DEFAULT current\_timestamp() | Konta izveides datums |

* 1. Scenārijs – elementa reģistrācija

Scenārija mērķis: demonstrēt, kā lietotājs reģistrē elementu spēlē *B.Iespējas*.

**Sākotnējais stāvoklis:**

* Spēlētājs vēl nav pieteicies spēlē.
* Sistēma atrodas sākumlapā, kur pieejama pieteikšanās forma.

**Turpmākie soļi:**

1. **Pieteikšanās:**
   * Spēlētājs ievada savu lietotājvārdu un paroli.
   * Sistēma validē datus un salīdzina ar datubāzē esošajiem ierakstiem.
   * Veiksmīgs gadījums: spēlētājs tiek autentificēts un pāradresēts uz spēles sākumlapu.
   * Neveiksmīgs gadījums: sistēma parāda kļūdas ziņojumu un piedāvā atkārtot mēģinājumu.
2. **Element reģistrācija:**
   * Spēlētājs izvēlas opciju "saglabāt elementu".
3. **Validācija:**
   * Pārbauda vai elementsa jau nav saglabāts.
4. **Datu apstrāde:**
   * Ja validācija ir veiksmīga, dati tiek saglabāti datubāzē tabulā "elementi".
   * Sistēma ģenerē unikālu resursa ID un reģistrē ievades datumu.
5. **Sistēmas atgriezeniskā saite:**
   * Neveiksmīga reģistrācija: sistēma atgriež formu ar kļūdas ziņojumu: “Elements jau ir saglabāts”.

**Beigu stāvoklis:**

* Lietotāja ievadītie dati ir veiksmīgi saglabāti datubāzē.
* Sistēma ir gatava pieņemt jaunus pieprasījumus.
  1. Scenārijs – resursu meklēšana bibloteka

Scenārija mērķis: demonstrēt, kā lietotājs meklē resursus, izmantojot *B.Iespējas*.

**Sākotnējais stāvoklis:**

* Spēlētājs vēl nav pieteicies sistēmā.

**Turpmākie soļi:**

1. **Pieteikšanās:**
   * Spēlētājs piesakās sistēmā, izmantojot savu lietotājvārdu un paroli.
2. **Meklēšana:**
   * Spēlētājs izvēlas sadaļu "Bibloteka".
   * Uzpiež us meklēšanas joslas un raksta elementa nosaukumu kuru vēlas atrast.
3. **Sistēmas reakcija:**
   * Sistēma veic meklēšanu datubāzē.
   * Atrasti ieraksti: tiek parādīta atbilstošā informācija.
   * Ieraksti netiek atrasti: tiek parādīts ziņojums: "Resursi netika atrasti!".

**Beigu stāvoklis:**

* Sistēma ir veikusi meklēšanas darbību un nodrošinājusi lietotāju ar rezultātiem.
  1. Scenārijs – jauna spēlētāja reģistrācija

Scenārija mērķis: demonstrēt, kā administrators pievieno jaunu spēlētāju sistēmai.

**Sākotnējais stāvoklis:**

* Administrators vēl nav pieteicies spēlē.

**Turpmākie soļi:**

1. **Pieteikšanās:**
   * Administrators piesakās sistēmā.
2. **Reģistrācija:**
   * Administrators izvēlas sadaļu "Reģistrēt jaunu spēlētāju".
   * Aizpilda formu ar šādu informāciju:
     + Spēlētāja lietotājvārds.
     + E-pasts.
     + Parole.
3. **Validācija un datu saglabāšana:**
   * Sistēma validē ievadīto informāciju un saglabā datus tabulā "users".

**Beigu stāvoklis:**

* Jaunais spēlētājs veiksmīgi pievienots sistēmai.
  1. Datu plūsmas process sistēmā
* **Lietotāja darbība:**

Sistēma saņem lietotāja ievadītos datus (piem., reģistrācija, resursu pievienošana).

* **Datu validācija:**

Sistēma validē ievades laukus un pārbauda datu precizitāti (piem., vai resursa daudzums ir pozitīvs skaitlis).

* **Datu saglabāšana:**

Apstiprinātie dati tiek saglabāti MySQL datu bāzē vai eksportēti uz CSV failu.

* **Datu atjaunināšana un dzēšana:**

Lietotāji var rediģēt vai dzēst esošos ierakstus, izmantojot lietotāja interfeisu.

1. testēšana
   1. Reģistrēšanās funkcijas testēšana

**Testa mērķis:** pārbaudīt lietotāja reģistrēšanos spēles **B.Iespējas** platformā.

**Testa gadījumi:**

4. Tabula reģistrēšanās funkcijas testēšana

|  |  |  |
| --- | --- | --- |
| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| Lietotājs aizpilda visus ievadlaukus, ievada lietotājvārdu, kas neeksistē datu bāzē, un abas ievadītās paroles ir vienādas. | Lietotājs tiek veiksmīgi reģistrēts un var piekļūt savam B.Iespējas profilam. | lietotājvārds:jauns@gmail.com  parole:123  atkārtota parole: 123  Izspildās gaidītais rezultāts. |
| Lietotājs mēģina reģistrēties, neaizpildot visus obligātos laukus. | Sistēma parāda kļūdas ziņojumu: "Visiem laukiem jābūt aizpildītiem!" | Lietotājvārds:jauns@gmail.com  parole:123  atkārtota parole:  Izspildās gaidītais rezultāts. |
| Lietotājs mēģina reģistrēties, izmantojot jau esošu lietotājvārdu. | Sistēma parāda kļūdas ziņojumu: "E-pasts jau tiek lietots!" | lietotājvārds:eksistējoš@gmail.com  parole:123  atkārtota parole: 123  Izspildās gaidītais rezultāts. |
| Lietotājs ievada dažādas paroles ievadlaukos. | Sistēma parāda kļūdas ziņojumu: "Paroles nesakrīt!" | Lietotājvārds:jauns@gmail.com  parole:123  atkārtota parole: 1234  Izspildās gaidītais rezultāts. |
| **Testa rezultāti** | **Ja lietotājs ievēro visus noteikumus, reģistrācija ir veiksmīga. Pretējā gadījumā sistēma sniedz precīzu kļūdas ziņojumu.** |  |

* 1. Pieteikšanās funkcijas testēšana

**Testa mērķis:** pārbaudīt lietotāja pieteikšanos **B.Iespējas** platformā.

**Testa gadījumi:**

5. Tabula pieteikšanās funkcijas testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs ievada pareizu lietotājvārdu un paroli. | Lietotājs veiksmīgi pieslēdzas spēlei. | Lietotājvārds:eksistējoš@gmail.com  parole:123  Izspildās gaidītais rezultāts. |
| Lietotājs ievada nepareizu paroli savam lietotājvārdam. | Sistēma parāda kļūdas ziņojumu: "E-pasts vai parole nav pareiza!". | Lietotājvārds:eksistējoš@gmail.com  parole:1233  Izspildās gaidītais rezultāts. |
| Lietotājs mēģina pieslēgties ar lietotājvārdu, kas neeksistē. | Sistēma parāda kļūdas ziņojumu: "E-pasts vai parole nav pareiza!". | Lietotājvārds:jauns@gmail.com  parole:123  Izspildās gaidītais rezultāts. |
| **Testa rezultāti** | **Sistēma korekti pārbauda lietotājvārdu un paroli, nodrošinot drošu piekļuvi.** |  |

* 1. Resursu meklēšanas funkcijas testēšana

**Testa mērķis:** pārbaudīt, kā lietotājs meklē resursus spēlē.

**Testa gadījumi:**

6. Tabula resursu meklēšanas funkcijas testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs meklē resursu, kas pastāv spēlē. | Lietotājam tiek parādīti pieejamie resursi. | Meklēšana:uguns  Izspildās gaidītais rezultāts. |
| Lietotājs meklē resursu, kas neeksistē. | Sistēma neko neizvada. | Meklēšana:vaiks  Izspildās gaidītais rezultāts. |
| **Testa rezultāti** | **Meklēšana darbojas pareizi, informējot lietotāju par rezultātiem.** |  |

* 1. Objekta veidošanas funkcijas testēšana

**Testa mērķis:** pārbaudīt, vai lietotājs var veiksmīgi veidot objektus no resursiem.

**Testa gadījumi:**

7. Tabula objekta veidošanas funkcijas testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs izmanto 2 elementus, lai izveidotu jaunu elementu. | Objekts tiek veiksmīgi izveidots un pievienots elementu sarakstam. | Elements1:uguns  Elements2:ūdens  Rezultāts:tvaiks  Izspildās gaidītais rezultāts. |

* 1. Elementa notīrīšanas funkcijas testēšana

**Testa mērķis:** pārbaudīt, vai lietotājs var veiksmīgi veidot notīrīt specifisku elementu.

**Testa gadījumi:**

8. Tabula Elementa notīrīšanas funkcijas testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs no spēkles laukuma ievelk elementu “Notīrīt šeit kastē”. | Elements tiek veiksmīgi notīrīts no spēles laukuma. | Lietotājam atlaižot elementu notīŗišanas laukā tas tiek veiksmīgi notīrīts no laukuma. |

* 1. Visu elementa notīrīšanas funkcijas testēšana

**Testa mērķis:** pārbaudīt, vai lietotājs var veiksmīgi veidot notīrīt visus elementus.

**Testa gadījumi:**

9. Tabula Visu elementa notīrīšanas funkcijas testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs vēlās atīrīt visu spēles lauku. | Lietotājs uzklikšķina uz notīrīšanas pogas kas apzīmēta ar slotiņu | Lietotājam noklikšķinot notīrīšanas pogu visi elementi tiek veiksmīgi notīrīts no laukuma. |

* 1. Elementa saglabāšanas funkcijas testēšana

**Testa mērķis:** pārbaudīt, vai lietotājs var veiksmīgi saglabāt specifisku elementu.

**Testa gadījumi:**

10. Tabula Elementa saglabāšanas funkcijas testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs no spēkles laukuma ievelk elementu “Saglabāt šeit” kastē. | Elements tiek veiksmīgi saglabāts datubāže un pazūd no spēles laukuma. | Lietotājam atlaižot elementu saglabāšanas laukā tas tiek veiksmīgi saglabāts datubāzē “*saved\_elements*” tabulā un notīrīts no laukuma. |

* 1. Elementa atrašanās biblotekā testēšana

**Testa mērķis:** pārbaudīt, vai lietotājam saglabājot elementu tas būs redzams biblotekas logā.

**Testa gadījumi:**

11. Tabula Elementa atrašanās biblotekā testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs no spēkles laukuma ievelk elementu “Saglabāt šeit” kastē. | Elements tiek veiksmīgi saglabāts datubāže un parādās biblotekā. | Lietotājam atlaižot elementu saglabāšanas laukā tas tiek veiksmīgi saglabāts datubāzē “*saved\_elements*” tabulā un parādās biblotekās logā. |

* 1. Atgriežanās funkcijas testēšana

**Testa mērķis:** pārbaudīt, vai lietotājs var veiksmīgi atgriezties uz iepriekšējo logu.

**Testa gadījumi:**

12. Tabula Atgriežanās funkcijas testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs vēlās atgriezties uz iepriekšējo logu noklikšķinot sarkano “X” pogu | Lietotājs uzklikšķina uz sarkano “X” pogu. | Lietotājam noklikšķinot atgriezties pogu tas tika aizvests uz iepriekšējo logu. |

* 1. Izrakstīšanās funkcijas testēšana

**Testa mērķis:** pārbaudīt, vai lietotājs var veiksmīgi izrakstīties no spēles profila.

**Testa gadījumi:**

13. Tabula Izrakstīšanās funkcijas testēšana

| **Situācija** | **Gaidītais rezultāts** | **Testa gaita** |
| --- | --- | --- |
| Lietotājs vēlās izrakstīties no sava profila | Lietotājs uzklikšķina uz “Logout” pogu. | Lietotājam noklikšķinot uz izrakstīšanās pogas tiek beigta esošā sesija. |

1. Lietotāja ceļvedis
   1. Ievads

Lietotāja ceļvedis ir būtiska dokumentācijas daļa, kas sniedz nepieciešamo informāciju par izstrādātā produkta lietošanu. Tā mērķis ir palīdzēt lietotājiem apgūt produkta funkcionalitātes un nodrošināt efektīvu produkta izmantošanu. Lietotāja ceļvedis ir noderīgs gan jaunajiem, gan pieredzējušajiem lietotājiem, nodrošinot labāku lietotāja pieredzi un veicinot produkta apguvi.

* 1. Produkta pieejamība (prototipa stadijā)

Tā kā izstrādātais produkts pašlaik atrodas prototipa stadijā, tas ir pieejams tikai lokāli vai izmantojot noteiktu piekļuves adresi. Prototipa versiju var apskatīt vietnē https://is-projekts.id.lv/, izmantojot piešķirto piekļuves informāciju.

Turpmākais lietotāja ceļvedis ir attiecināms uz šo prototipa versiju. Produkta pieejamība un funkcionalitāte var mainīties, attīstot to tālāk līdz publiskai versijai.

* 1. Pieteikšanās logs
     1. ****Vispārējs apraksts****

Atverot *web* produktu, lietotājs tiek novirzīts uz pierakstīšanās lapu. Šajā lapā ir izvietota aizpildforma ar ievadlaukumiem, kur reģistrētie lietotāji var ievadīt savu e-pasta adresi un paroli, lai piekļūtu savam profilam un iespējotu turpmāku darbību (skat. 26. att.).

Ja lietotājs vēl nav reģistrēts, viņam ir iespēja doties uz reģistrācijas lapu, izmantojot teksta pogu "Reģistrēties".

A screenshot of a computer

AI-generated content may be incorrect.

30. att. Neaizpildītas pieteikšanās lapas izskats.

* + 1. Darbība

**Pierakstīšanās**

1. Reģistrēts lietotājs aizpilda ievadlaukos savu reģistrēto e-pasta adresi un paroli.
2. Pēc informācijas aizpildīšanas lietotājs nospiež pogu "Pieteikties", lai apstiprinātu pierakstīšanās informāciju un piekļūtu savam profilam (skat. 27. att.).

A screenshot of a login form

AI-generated content may be incorrect.

31. att. Aizpildītas pierakstīšanās lapas izskats.

**Pāreja uz reģistrēšanos**

1. Ja lietotājs vēl nav reģistrēts, viņš nospiež uz teksta pogas **"Reģistrēties"**, kas novirza uz reģistrācijas lapu, kur iespējams izveidot jaunu profilu (skat. 28**. att.**).

A screenshot of a computer

AI-generated content may be incorrect.

32. att. Pāreja uz reģistrēšanās lapu.

* 1. Reģistrēšanās lapa
     1. ****Vispārējs apraksts****

Reģistrēšanās lapā ir aizpildforma, kurā nereģistrēti lietotāji, izmantojot iepriekš nereģistrētu e-pasta adresi un izvēloties atbilstošu paroli (ievadot to divas reizes), var reģistrēties produkta izmantošanai (skat. 29**. att.**).

Ja lietotājs jau ir reģistrēts, viņam jāizmanto teksta poga **"Login"**, lai pārietu uz pierakstīšanās lapu.

A screenshot of a computer

AI-generated content may be incorrect.

33. att. Neaizpildītas reģistrēšanās lapas izskats.

* + 1. Darbība

**Reģistrēšanās**

1. Nereģistrēts lietotājs aizpilda ievadlaukus:
   * **E-pasta adresi**, kas ir unikāla un nav reģistrēta sistēmā.
   * **Paroli**.
   * **Atkārtotu paroli**, lai pārliecinātos par precīzu ievadi.
2. Parolēm jābūt vienādām.
3. Pēc ievadlauku aizpildīšanas lietotājs nospiež pogu **"Register"**, lai apstiprinātu reģistrēšanās informāciju un izveidotu jaunu profilu (skat. 30**. att.).**

A screenshot of a login form

AI-generated content may be incorrect.

34. att. Aizpildītas reģistrēšanās lapas izskats.

**Pāreja uz pierakstīšanos**

Ja lietotājs jau ir reģistrējies, viņš nospiež teksta pogu **"Login"**, lai pārietu uz pierakstīšanās lapu (skat. 31**. att.**).

A screenshot of a computer

AI-generated content may be incorrect.

35. att. Pāreja uz pierakstīšanās lapu.

* 1. Sākumlapa
     1. Vispārējs apraksts

Pēc veiksmīgas pierakstīšanās vai reģistrēšanās lietotājs tiek novirzīts uz sākumlapu jeb **homepage logu**. Sākumlapa kalpo kā galvenais izvēlnes logs, no kura lietotājs var piekļūt dažādām produkta sadaļām un funkcionalitātēm (skat. 32. att**.**).

Sākumlapā ir redzams satura virsraksts **"B.Iespējas"**, kā arī četras galvenās pogas:

1. **"Jauna Spēle"** – nospiežot šo pogu, lietotājs uzsāk jaunu spēli.
2. **"Bibliotēka"** – nospiežot šo pogu, lietotājs var apskatīt bibliotēku, kurā ir visi iepriekš saglabātie elementi.
3. **"Log out"** – nospiežot šo pogu, lietotājs tiek izrakstīts no sava profila un novirzīts uz pierakstīšanās lapu.

A screenshot of a computer

AI-generated content may be incorrect.

36. att. Sākumlapas izskats ar pieejamajām pogām.

* + 1. Darbība

**Jauna spēle**

* Nospiežot pogu **"Jauna Spēle"**, lietotājs tiek novirzīts uz spēles logu, kur tiek uzsākta jauna spēle no sākuma.

**Bibliotēka**

* Nospiežot pogu **"Bibliotēka"**, lietotājs tiek novirzīts uz bibliotēkas logu, kur var apskatīt visus iepriekš saglabātos elementus.

**Iziet**

* Nospiežot pogu **"Log out"**, lietotājs tiek izrakstīts no sava profila un novirzīts uz pierakstīšanās lapu.
  1. Bibliotēkas logs
     1. Vispārējs apraksts

Bibliotēkas logs ir sadaļa, kurā lietotājs var apskatīt visus iepriekš saglabātos elementus. Šis logs ļauj lietotājam pārlūkot un meklēt saglabātos elementus. (skat. 33**. att.**)

Bibliotēkas logā ir redzams satura virsraksts **"Bibliotēka"**, meklēšanas josla un saraksts ar elementiem. Lietotājam ir arī iespēja atgriezties uz sākumlapu, izmantojot pogu **"X"**, kas atrodas ekrāna augšējā kreisajā stūrī.

A screenshot of a computer

AI-generated content may be incorrect.

37. att. Bibliotēkas lapas izskats bez elementiem.

* + 1. ****Darbība****

**Elementu saraksta pārlūkošana**

* Kad lietotājs atver bibliotēkas logu, tajā tiek parādīts saraksts ar visiem iepriekš saglabātajiem elementiem. (Skat. 34**. att**.)

A screenshot of a computer

AI-generated content may be incorrect.

38. att. Biblotekas loga elementu saraksts

**Meklēšanas funkcija**

* Lietotājs var izmantot meklēšanas joslu, lai atrastu konkrētu elementu pēc tā nosaukuma.
* Ievadot tekstu meklēšanas joslā, saraksts tiek filtrēts un tiek parādīti tikai tie elementi, kuru nosaukums atbilst ievadītajam tekstam. (Skat.35**. att.**)

A screenshot of a computer

AI-generated content may be incorrect.

39. att. Meklēšanas josla darbībā

**Atgriešanās uz sākumlapu**

* Lietotājs var nospiest pogu **"X"**, lai atgrieztos uz sākumlapu. (Skat. 36**. att.**, kur redzama poga "X".)

A screenshot of a computer

AI-generated content may be incorrect.

40. att. Atgriezsties poga biblotekas logā

* 1. Spēles logs
     1. ****Vispārējs apraksts****

Spēles logs ir galvenā vieta, kur lietotājs var veidot jaunus elementus, kombinējot jau esošos elementus. Lietotājs šajā logā var vilkt elementus no elementu saraksta un novietot tos spēles laukā, lai kombinētu un iegūtu jaunus elementus. (Skat. 37. att.)

Spēles logā ir pieejamas trīs galvenās zonas un darbības pogas:

1. **Elementu saraksts** – atrodas labajā pusē, kur lietotājs var izvēlēties pieejamos elementus un meklēt tos ar meklēšanas funkciju.
2. **Spēles laukums** – galvenā zona, kur lietotājs var novietot un kombinēt elementus.
3. **Funkciju pogas** – pogas atrodas spēles laukuma malās:
   * **"Notīrīt"** – lai no spēles laukuma izdzēstu elementu.
   * **“Notīrīt visu”** – **lai no spēles laukuma izdzēstu visus novietotos elementus.**
   * **"Saglabāt šeit"** – lai saglabātu izveidotos elementus bibliotēkā.
   * **"X"** – lai atgrieztos sākumlapā.

A screenshot of a computer

Description automatically generated

41. att. Spēles loga izskats.

* + 1. Darbība

**Elementu vilkšana un novietošana**

1. Lietotājs izvēlas elementu no saraksta, kas atrodas spēles loga labajā pusē.
2. Izvēlēto elementu var vilkt un novietot spēles laukuma jebkurā vietā. (Skat. 38**. att**.)

A screenshot of a computer

Description automatically generated

42. att. Elements spēles laukumā

**Elementu kombinēšana**

1. Lai izveidotu jaunu elementu, lietotājs novieto divus elementus pārklājot vienu otram.
2. Abi elementi pazūd un to vietā parādās jaunais elements. (Skat. 39. att.)

A screenshot of a computer

Description automatically generated

43. att. Jauns elements spēles laukumā

**Elementu saglabāšana**

1. Kad lietotājs ir izveidojis jaunu elementu un vēlas to saglabāt, viņš var vilkt šo elementu uz pogu "Saglabāt šeit", kas atrodas spēles laukuma malā.
2. Elements tiek pievienots bibliotēkai un lietotājs saņem apstiprinājumu. (Skat. 40. att., kur redzama elementa saglabāšana.)

A screenshot of a computer

Description automatically generated

44. att. Saglabāt šeit lauks

**Meklēšanas funkcija**

* Lietotājs var izmantot meklēšanas joslu, kurš atrodās elementu sarakstā, lai atrastu konkrētu elementu pēc tā nosaukuma.
* Ievadot tekstu meklēšanas joslā, saraksts tiek filtrēts un tiek parādīti tikai tie elementi, kuru nosaukums atbilst ievadītajam tekstam. (Skat.41**. att.)**

A screenshot of a computer

Description automatically generated

45. att. Meklēšanas josla darbībā elementu joslā

**Laukuma notīrīšana**

1. Lai notīrītu visus elementus no spēles laukuma, lietotājs var nospiest pogu **"Notīrīt"**.
2. Nospiežot pogu, spēles laukums tiek pilnībā iztukšots un lietotājs var sākt veidot jaunus elementus. (Skat. 41**. att.**, kur redzama funkcija "Notīrīt".)

A screenshot of a computer

Description automatically generated

46. att. Notīrīt poga

**Notīrīt laukums**

1. Lai notīrītu specifisku elementu no spēles laukuma, lietotājs to var ievilkt “Notīrīt šeit” pogu kura atrodās apakšējā kreisajā stūrī.
2. Ievelkot elementu tas tiek nodzēsts no spēles laukuma. (Skat. 42**. att.**, kur redzama funkcija "Notīrīt šeit".)

A screenshot of a computer

Description automatically generated

47. att. Notīrīt šeit lauks

**Atgriešanās uz sākumlapu**

1. Lai atgrieztos sākumlapā, lietotājs nospiež pogu **"Atgriezties"**, kas atrodas augšējā kreisajā stūrī.
2. Nospiežot šo pogu, lietotājs tiek novirzīts uz sākumlapas logu. (Skat.43**. att.**, kur redzama funkcija "Atgriezties".)

A screenshot of a computer

Description automatically generated

48. att. Atgriezties poga spēles loga

Secinājumi un priekšlikumi

1. Pirms projekta uzsākšanas tika analizētas līdzīgas darba uzdevumu vadības sistēmas, identificējot galvenās problēmas, ar kurām varētu saskarties lietotāji. Šīs analīzes rezultātā tika izveidotas vadlīnijas, kas palīdzēja izvairīties no līdzīgu problēmu atkārtošanas, nodrošinot ērtu un efektīvu lietotāja pieredzi.
2. Izprasti būtiskākie produkta darbības principi un tā funkcionalitātes, kas palīdzēja precīzi plānot darba uzdevumus un sadalīt izstrādes procesu posmos, veicinot efektīvu darba organizāciju.
3. Pirms kodēšanas uzsākšanas tika izveidoti produkta maketi (*wireframe*) un saskaņoti ar pasniedzēju. Maketu izstrāde palīdzēja vizualizēt produkta izskatu un funkcionalitāti, ļaujot sekot iepriekš izstrādātajam plānam. Šis posms prasīja radošu domāšanu un izpratni par dizainiskajiem un funkcionālajiem risinājumiem.
4. Datu glabāšanai tika izvēlēta uzticama un viegli lietojama datu bāzes pārvaldības programmatūra ***HeidiSQL***, kas nodrošināja ērtu datu glabāšanu un iespēju tos viegli rediģēt, kā arī nodrošināja ātru savienojumu ar produkta kodu.
5. Produkta izstrādei tika izvēlēts populārs un efektīvs kodēšanas redaktors ***Visual Studio Code***, kas nodrošināja ātru un ērtu koda rakstīšanu un rediģēšanu, izmantojot pieejamos spraudņus un papildiespējas.
6. Tika izveidots pilnībā funkcionējošs un prasībām atbilstošs produkts, kas ietver visas paredzētās sadaļas un logus ar nepieciešamajām funkcionalitātēm. Lai gan produkts ir prototipa stadijā, tas pilnībā atbilst definētajām prasībām un ir gatavs turpmākai attīstībai.
7. Produkta izstrādes laikā tika veikta regulāra testēšana, lai nodrošinātu atbilstību prasībām un funkcionalitātes pareizību. Pēc izstrādes tika veikta gala testēšana, kas dokumentēta gala dokumentācijā, un tās rezultāti apliecina, ka produkts veiksmīgi izturējis visus testus.
8. Dokumentācija tika izveidota atbilstoši izvirzītajām prasībām, nodrošinot pilnvērtīgu un kvalitatīvu informāciju par izstrādāto produktu. Tika iegūta vērtīga pieredze reālas dokumentācijas veidošanā, kas noderēs turpmākajos profesionālajos projektos.
9. Kopumā ir veiksmīgi izstrādāts kvalifikācijas darba produkts un dokumentācija. Projekta izstrādes laikā tika iegūta vērtīga pieredze, kas simulēja programmētāja reālo darba procesu – sākot no prasību analīzes un izstrādes plānošanas līdz produkta izstrādei, testēšanai un dokumentācijas veidošanai.

Izmantoto avotu saraksts

1. **MindOnMap**, Wireframe veidošanas pamācība, 2024. [Tiešsaiste]. Pieejams: <https://web.mindonmap.com/files> [skatīts 25.09.2024.]
2. **Javatpoint**, HTML ievads – pamati, 2024. [Tiešsaiste]. Pieejams: <https://www.javatpoint.com/what-is-html> [skatīts 20.10.2024.]
3. **MDN Web Docs**, CSS pamācības un piemēri, 2024. [Tiešsaiste]. Pieejams: <https://developer.mozilla.org/en/docs/Web/CSS> [skatīts 10.10.2024.]
4. **Microsoft**, Visual Studio Code lietošanas dokumentācija, 2023. [Tiešsaiste]. Pieejams: <https://learn.microsoft.com/en-us/powershell/scripting/dev-cross-plat/vscode/using-vscode-for-debugging-compiled-cmdlets?view=powershell-7.4> [skatīts 18.10.2024.]
5. **Stack Overflow**, Programmēšanas jautājumi un atbildes. [Tiešsaiste]. Pieejams: <https://stackoverflow.com/> [skatīts 22.10.2024.]
6. **Traversy Media**, Web izstrādes pamācības, 2024. [Tiešsaiste]. Pieejams: <https://www.youtube.com/user/TechGuyWeb> [skatīts 16.10.2024.]
7. **W3Schools**, PHP pamācība, 2024. [Tiešsaiste]. Pieejams: <https://www.w3schools.com/php/> [skatīts 25.10.2024.]
8. **YouTube**, JavaScript interaktīvas funkcionalitātes izveide, 2024. [Tiešsaiste]. Pieejams:<https://www.youtube.com/watch?v=PkZNo7MFNFg> [skatīts 21.10.2024.]
9. **OpenAI Platform** mākslīga intelekta pielitošana, 2024. [Tiešsaiste]. Pieejams: <https://platform.openai.com/docs/api-reference/introduction> [skatīts 13.10.2024.]
10. **HeidiSQL**, Datubāzu pārvaldības rīka apraksts. [Tiešsaiste]. Pieejams: <https://www.heidisql.com/> [skatīts 15.10.2024.]
11. **Dave**, XAMPP Server Overview, 2023. [Tiešsaiste]. Pieejams: <https://cloudfoundation.com/blog/what-is-xampp/> [skatīts 18.10.2024.]
12. pielikums **game.js**

document.addEventListener("DOMContentLoaded", () => {

const elementList = document.getElementById("element-list");

const resizeHandle = document.createElement("div");

resizeHandle.id = "resize-handle";

document.body.appendChild(resizeHandle);

const gameContainer = document.getElementById("game-container");

const saveArea = document.getElementById("save-area");

const clearArea = document.getElementById("clear-area");

const searchBar = document.querySelector(".search-bar");

const clearAllButton = document.getElementById("clear-all-button");

let dragOffsetX = 0;

let dragOffsetY = 0;

let isResizing = false;

// Sānu joslas izmēru maiņa

resizeHandle.addEventListener("mousedown", (event) => {

isResizing = true;

document.body.style.cursor = "ew-resize";

event.preventDefault();

});

document.addEventListener("mousemove", (event) => {

if (isResizing) {

const newWidth = Math.max(window.innerWidth - event.clientX, 100); // Minimālais platums ir 100px

elementList.style.width = `${newWidth}px`;

resizeHandle.style.right = `${newWidth}px`;

}

});

document.addEventListener("mouseup", () => {

if (isResizing) {

isResizing = false;

document.body.style.cursor = "";

}

});

// Meklēšanas joslas ievade

searchBar.addEventListener("input", (event) => {

const query = event.target.value.toLowerCase();

const elements = elementList.querySelectorAll(".draggable-button");

elements.forEach((element) => {

const elementText = element.textContent.toLowerCase();

element.style.display = elementText.includes(query) ? "" : "none";

});

});

// Poga "Notīrīt visu"

clearAllButton.addEventListener("click", () => {

const elements = gameContainer.querySelectorAll(".game-element");

elements.forEach((element) => element.remove());

alert("Visi elementi notīrīti!");

});

// Datu ielāde no servera un elementu saraksta veidošana

fetch("fetch\_elements.php")

.then((response) => response.json())

.then((elements) => {

elements.forEach((element) => {

const button = document.createElement("div");

button.classList.add("draggable-button");

button.draggable = true;

button.dataset.element = element.name;

button.textContent = element.name;

// Notikums velkšanai no elementu saraksta

button.addEventListener("dragstart", (event) => {

const rect = event.target.getBoundingClientRect();

dragOffsetX = event.clientX - rect.left;

dragOffsetY = event.clientY - rect.top;

event.dataTransfer.setData("text/plain", event.target.dataset.element);

event.dataTransfer.setData("source", "element-list");

});

elementList.appendChild(button);

});

})

.catch((error) => console.error("Kļūda, ielādējot elementus:", error));

// Velkamo elementu apstrāde spēles laukā

gameContainer.addEventListener("dragover", (event) => event.preventDefault());

gameContainer.addEventListener("drop", (event) => {

event.preventDefault();

const draggedElement = document.querySelector(".dragging");

const elementType = event.dataTransfer.getData("text/plain");

const source = event.dataTransfer.getData("source");

// Ja elements tiek vilkts no saraksta

if (source === "element-list") {

const newX = event.clientX - dragOffsetX - gameContainer.offsetLeft;

const newY = event.clientY - dragOffsetY - gameContainer.offsetTop;

const newElement = createGameElement(elementType);

newElement.style.left = `${newX}px`;

newElement.style.top = `${newY}px`;

gameContainer.appendChild(newElement);

checkForCombination(newElement);

}

// Ja elementi tiek apvienoti spēles laukumā

if (source === "game-container" && draggedElement) {

const newX = event.clientX - dragOffsetX - gameContainer.offsetLeft;

const newY = event.clientY - dragOffsetY - gameContainer.offsetTop;

draggedElement.style.left = `${newX}px`;

draggedElement.style.top = `${newY}px`;

checkForCombination(draggedElement);

}

});

// Elementu apvienošana spēles laukumā

function checkForCombination(draggedElement) {

const dropTargets = Array.from(gameContainer.querySelectorAll(".game-element")).filter(

(el) => el !== draggedElement

);

let overlappingTarget = null;

dropTargets.forEach((target) => {

const targetRect = target.getBoundingClientRect();

const draggedRect = draggedElement.getBoundingClientRect();

const isOverlapping =

targetRect.left < draggedRect.right &&

targetRect.right > draggedRect.left &&

targetRect.top < draggedRect.bottom &&

targetRect.bottom > draggedRect.top;

if (isOverlapping) {

overlappingTarget = target;

}

});

if (overlappingTarget) {

const element1 = draggedElement.dataset.element;

const element2 = overlappingTarget.dataset.element;

const x = (parseFloat(draggedElement.style.left) + parseFloat(overlappingTarget.style.left)) / 2;

const y = (parseFloat(draggedElement.style.top) + parseFloat(overlappingTarget.style.top)) / 2;

draggedElement.remove();

overlappingTarget.remove();

fetch("combine\_elements.php", {

method: "POST",

headers: {

"Content-Type": "application/x-www-form-urlencoded",

},

body: `element1=${encodeURIComponent(element1)}&element2=${encodeURIComponent(element2)}`,

})

.then((response) => response.json())

.then((data) => {

if (data.status === "success" || data.status === "exists") {

const newElement = data.newElement;

const existingElement = Array.from(elementList.children).find(

(child) => child.textContent === newElement

);

if (!existingElement) {

const button = document.createElement("div");

button.classList.add("draggable-button");

button.draggable = true;

button.dataset.element = newElement;

button.textContent = newElement;

button.addEventListener("dragstart", (event) => {

const rect = event.target.getBoundingClientRect();

dragOffsetX = event.clientX - rect.left;

dragOffsetY = event.clientY - rect.top;

event.dataTransfer.setData("text/plain", event.target.dataset.element);

event.dataTransfer.setData("source", "element-list");

});

elementList.appendChild(button);

}

const gameElement = createGameElement(newElement);

gameElement.style.left = `${x}px`;

gameElement.style.top = `${y}px`;

gameContainer.appendChild(gameElement);

} else {

alert("Neizdevās apvienot elementus: " + data.message);

}

})

.catch((error) => console.error("Kļūda, apvienojot elementus:", error));

}

}

// Saglabāšanas laukuma funkcionalitāte

saveArea.addEventListener("dragover", (event) => event.preventDefault());

saveArea.addEventListener("drop", (event) => {

event.preventDefault();

const draggedElement = document.querySelector(".dragging");

if (draggedElement && draggedElement.parentNode === gameContainer) {

const elementType = draggedElement.dataset.element;

fetch("save\_element.php", {

method: "POST",

headers: {

"Content-Type": "application/x-www-form-urlencoded",

},

body: `name=${encodeURIComponent(elementType)}`

})

.then((response) => response.json())

.then((data) => {

if (data.status === "success") {

alert("Elements veiksmīgi saglabāts!");

draggedElement.remove();

} else {

alert("Neizdevās saglabāt elementu.");

}

})

.catch((error) => console.error("Kļūda, saglabājot elementu:", error));

}

});

fetch('fetch\_saved\_elements.php')

.then(response => response.json())

.then(data => {

console.log("Fetched elements:", data);

if (Array.isArray(data)) {

data.forEach(element => {

//Prevent duplicates by checking if element already exists visually

const alreadyExists = gameContainer.querySelector(`[data-element="${element.name}"]`);

if (!alreadyExists) {

const el = createGameElement(element.name);

el.style.left = `${Math.random() \* 300}px`;

el.style.top = `${Math.random() \* 300}px`;

}

});

} else {

console.error("Failed to load elements:", data);

}

})

.catch(err => console.error("Error loading elements:", err));

// Dzēšanas laukuma funkcionalitāte

clearArea.addEventListener("dragover", (event) => event.preventDefault());

clearArea.addEventListener("drop", (event) => {

event.preventDefault();

const draggedElement = document.querySelector(".dragging");

if (draggedElement && draggedElement.parentNode === gameContainer) {

draggedElement.remove();

}

});

// Palīgfunkcija spēles elementa izveidei

function createGameElement(type) {

const element = document.createElement("div");

element.classList.add("game-element");

element.dataset.element = type;

element.textContent = type;

element.style.position = "absolute";

element.draggable = true;

element.addEventListener("dragstart", (event) => {

const rect = event.target.getBoundingClientRect();

dragOffsetX = event.clientX - rect.left;

dragOffsetY = event.clientY - rect.top;

event.dataTransfer.setData("text/plain", event.target.dataset.element);

event.dataTransfer.setData("source", "game-container");

element.classList.add("dragging");

});

element.addEventListener("dragend", () => {

element.classList.remove("dragging");

});

return element;

}

});

// Funkcija, lai atgrieztos uz sākumlapu

function goBack() {

window.location.href = "homepage.html"; // Pielāgot ceļu uz sākumlapu

}

1. pielikums **homepage.js**

function navigateToGame() {

window.location.href = 'game.html'; // Pāradresē uz spēles lapu

}

function navigateToLibrary() {

window.location.href = 'library.html'; // Pāradresē uz bibliotēkas lapu

}

function logout() {

window.location.href = 'logout.php'; // Izsauc izrakstīšanās skriptu

}

1. pielikums **index.js**

function toggleForm(formId) {

const registrationForm = document.getElementById('registration');

const loginForm = document.getElementById('login');

// Rāda vai paslēpj reģistrācijas un pieteikšanās formu atkarībā no izvēlētās formas

registrationForm.style.display = formId === 'registration' ? 'block' : 'none';

loginForm.style.display = formId === 'login' ? 'block' : 'none';

}

function navigateToHome() {

window.location.href = 'homepage.html'; // Pāradresē uz sākumlapu

}

document.addEventListener("DOMContentLoaded", () => {

const registrationForm = document.querySelector("#registration form");

const passwordInput = document.getElementById("password");

const confirmPasswordInput = document.getElementById("confirm-password");

const lengthRequirement = document.getElementById("length");

const uppercaseRequirement = document.getElementById("uppercase");

passwordInput.addEventListener("input", () => {

const password = passwordInput.value;

// Length requirement

if (password.length >= 8) {

lengthRequirement.classList.remove("invalid");

lengthRequirement.classList.add("valid");

} else {

lengthRequirement.classList.add("invalid");

lengthRequirement.classList.remove("valid");

}

// Uppercase requirement

if (/[A-Z]/.test(password)) {

uppercaseRequirement.classList.remove("invalid");

uppercaseRequirement.classList.add("valid");

} else {

uppercaseRequirement.classList.add("invalid");

uppercaseRequirement.classList.remove("valid");

}

});

registrationForm.addEventListener("submit", function (e) {

const password = passwordInput.value;

const confirmPassword = confirmPasswordInput.value;

const isValidLength = password.length >= 8;

const hasUppercase = /[A-Z]/.test(password);

if (!isValidLength || !hasUppercase) {

alert("Parole neatbilst prasībām.");

e.preventDefault();

return;

}

if (password !== confirmPassword) {

alert("Paroles nesakrīt.");

e.preventDefault();

}

});

});

1. pielikums **library.js**

document.addEventListener("DOMContentLoaded", () => {

const elementsContainer = document.querySelector(".elements");

const searchBar = document.querySelector(".search-bar");

// Ielādē saglabātos elementus no servera

fetch("fetch\_library\_elements.php")

.then(response => response.json())

.then(elements => {

console.log("Library elements fetched:", elements);

elements.forEach(element => {

const el = document.createElement("div");

el.classList.add("element-item");

el.textContent = element.name;

elementsContainer.appendChild(el);

});

})

.catch(error => console.error("Kļūda, ielādējot bibliotēkas elementus:", error));

// Meklēšana

searchBar.addEventListener("input", (event) => {

const query = event.target.value.toLowerCase();

const elementItems = elementsContainer.querySelectorAll(".element-item");

elementItems.forEach((item) => {

const itemText = item.textContent.toLowerCase();

item.style.display = itemText.includes(query) ? "" : "none";

});

});

});

function goBack() {

window.location.href = 'homepage.html';

}

1. pielikums **game.css**

/\* Vispārīgi ķermeņa stila iestatījumi \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

height: 100vh;

overflow: hidden;

}

/\* Fona video stila iestatījumi \*/

video#background-video {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

object-fit: cover;

z-index: -1;

}

/\* Spēles konteinera stila iestatījumi \*/

.game-container {

position: relative;

width: 100%;

height: 100%;

overflow: hidden;

}

/\* Atpakaļ pogas stila iestatījumi \*/

.button {

position: absolute;

top: 1rem;

left: 1rem;

padding: 0.5rem 1rem;

font-size: 18px;

background-color: #ff2f2f52;

color: #FFFFFF;

border: none;

border-radius: 4px;

cursor: pointer;

z-index: 10;

}

.button:hover {

background-color: #ff5050;

}

/\* Dzēšanas un saglabāšanas laukuma stila iestatījumi \*/

.clear-area, .save-area {

position: absolute;

bottom: 1rem;

width: 150px;

height: 50px;

color: #fff;

border: 2px dashed #fff;

border-radius: 4px;

text-align: center;

line-height: 50px;

font-size: 16px;

cursor: pointer;

z-index: 10;

}

.clear-area {

left: 1rem;

background-color: rgba(211, 211, 211, 0.24); /\* Gaiši pelēks \*/

}

.clear-area:hover {

background-color: rgba(169, 169, 169, 1); /\* Tumši pelēks, ja virsū novieto kursoru \*/

}

.save-area {

right: 22%;

background-color: rgba(211, 211, 211, 0.24); /\* Gaiši rozā \*/

}

.save-area:hover {

background-color: rgba(169, 169, 169, 1); /\* Spilgti rozā, ja virsū novieto kursoru \*/

}

/\* Spēles elementi \*/

.game-element {

padding: 0.2rem 0.4rem;

text-align: center;

border: 1px solid #ccc;

border-radius: 4px;

background-color: rgba(255, 255, 255, 0.8);

color: #000;

cursor: move;

position: absolute;

white-space: nowrap;

width: fit-content;

font-size: 1rem;

}

/\* Elementu saraksta stila iestatījumi \*/

.element-list {

position: absolute;

top: 0;

right: 0;

width: 20%;

height: 100%;

background-color: rgba(255, 255, 255, 0.2);

border-left: 2px solid #FFFFFF;

padding: 1rem;

box-sizing: border-box;

overflow-y: auto;

color: #FFFFFF;

z-index: 5;

}

/\* Meklēšanas joslas stila iestatījumi \*/

.search-bar {

width: 100%; /\* Nodrošina, ka tā paliek saraksta platumā \*/

padding: 0.5rem;

margin-bottom: 1rem;

border: 1px solid #FFFFFF;

border-radius: 4px;

background-color: rgba(255, 255, 255, 0.2);

color: #FFFFFF;

box-sizing: border-box; /\* Iekļauj atstarpes platuma aprēķinā \*/

overflow: hidden; /\* Novērš pārlieku lielu saturu \*/

}

.search-bar::placeholder {

color: #FFFFFF; /\* Viettura teksta krāsa \*/

}

/\* Velkamo pogu stila iestatījumi \*/

.draggable-button {

padding: 0.2rem 0.4rem;

margin: 0.5rem 0;

text-align: center;

border: 1px solid #FFFFFF;

border-radius: 4px;

background-color: rgba(255, 255, 255, 0.2);

color: #FFFFFF;

cursor: grab;

display: inline-block;

font-size: 1rem;

white-space: nowrap; /\* Novērš teksta pārnešanu nākamajā rindā \*/

width: fit-content; /\* Automātiski pielāgo platumu tekstam \*/

}

.draggable-button:active {

cursor: grabbing;

}

/\* Poga "Notīrīt visu" stila iestatījumi \*/

.clear-all-button {

position: absolute;

bottom: 1rem;

left: calc(2rem + 150px + 1rem);

transform: translateX(-50%);

width: 50px;

height: 50px;

background: none; /\* Bez fona attēla \*/

border: none;

cursor: pointer;

z-index: 10;

}

/\* Pogas attēla stila iestatījumi \*/

.button-image {

width: 120%;

height: 100%;

filter: brightness(0) invert(1); /\* Nodrošina, ka attēls ir balts \*/

}

1. pielikums **homepage.css**

/\* Ķermeņa stils \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

height: 100vh;

overflow: hidden;

display: flex;

justify-content: center;

align-items: center;

background-color: #000;

}

/\* Fona video \*/

video#background-video {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

object-fit: cover;

z-index: -1;

}

/\* Sākumlapas konteiners \*/

.homepage {

width: 100%;

max-width: 400px;

background-color: rgba(255, 255, 255, 0.2);

color: #FFFFFF;

border: 2px solid #FFFFFF;

border-radius: 8px;

padding: 2rem;

text-align: center;

}

/\* Virsraksts \*/

h1 {

font-size: clamp(1rem, 5vmin, 3rem);

font-family: sans-serif;

color: hsl(0, 0%, 100%);

text-align: center;

margin-top: 30px;

margin-bottom: 60px;

}

.underlined {

/\* background: red; \*/

position: relative;

}

.underline-a:after {

content: '';

position: absolute;

top: 120%;

width: 180%;

aspect-ratio: 3 / 1;

left: 50%;

transform: translate(-50%, 0);

border-radius: 50%;

border: 6px solid hsl(202, 82%, 56%);

clip-path: polygon(0 0, 50% 50%, 100% 0);

}

/\* Poga \*/

button {

width: 100%;

padding: 0.7rem;

font-size: 1rem;

color: #FFFFFF;

background-color: #007bff;

border: none;

border-radius: 20px;

cursor: pointer;

margin-bottom: 1rem;

}

/\* Pogas hover efekts \*/

button:hover {

background-color: #0056b3;

}

1. pielikums **index.css**

/\* Pamatstili ķermenim un izvietojumam \*/

body {

font-family: Arial, sans-serif;

margin: 0;

padding: 0;

height: 100vh;

overflow: hidden;

display: flex;

justify-content: center;

align-items: center;

background-color: #000;

}

/\* Fona video stils \*/

video#background-video {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

object-fit: cover;

z-index: -1;

}

/\* Konteinera un lapas stils \*/

.container, .homepage {

width: 100%;

max-width: 400px;

background-color: rgba(255, 255, 255, 0.2);

color: #FFFFFF;

border: 2px solid #FFFFFF;

border-radius: 8px;

padding: 2rem;

box-sizing: border-box;

text-align: center;

}

h1 {

font-size: clamp(1rem, 5vmin, 3rem);

font-family: sans-serif;

color: hsl(0, 0%, 100%);

text-align: center;

margin-bottom: 40px;

}

.underlined {

/\* background: red; \*/

position: relative;

}

.underline-a:after {

content: '';

position: absolute;

top: 120%;

width: 180%;

aspect-ratio: 3 / 1;

left: 50%;

transform: translate(-50%, 0);

border-radius: 50%;

border: 6px solid hsl(202, 82%, 56%);

clip-path: polygon(0 0, 50% 50%, 100% 0);

}

/\* Teksta un ievades lauku stils \*/

label {

display: block;

margin-bottom: 0.5rem;

color: #FFFFFF;

}

input {

width: 100%;

padding: 0.5rem;

margin-bottom: 1rem;

font-size: 1rem;

border: 1px solid #FFFFFF;

border-radius: 4px;

background-color: rgba(255, 255, 255, 0.2);

color: #FFFFFF;

}

input::placeholder {

color: #FFFFFF;

}

/\* Pogas stils \*/

button {

width: 100%;

padding: 0.7rem;

font-size: 1rem;

color: #FFFFFF;

background-color: #007bff;

border: none;

border-radius: 20px;

cursor: pointer;

}

button:hover {

background-color: #0056b3;

}

/\* Papildteksta un saišu stils \*/

.text {

margin-top: 1rem;

font-size: 1.2rem;

}

.text a {

color: #FFFFFF;

text-decoration: underline;

}

.text a:hover {

color: #ccc;

}

#password-requirements {

display: flex;

gap: 0.5rem;

margin-bottom: 1rem;

flex-wrap: wrap;

justify-content: center;

}

.requirement {

padding: 0.3rem 0.6rem;

border-radius: 20px;

font-size: 0.85rem;

border: 1px solid;

white-space: nowrap;

}

.requirement.invalid {

color: #ff4d4d;

border-color: #ff4d4d;

background-color: rgba(255, 77, 77, 0.1);

}

.requirement.valid {

color: #00ff7f;

border-color: #00ff7f;

background-color: rgba(0, 255, 127, 0.1);

}

1. pielikums **library.css**

/\* Vispārīgs ķermeņa stils \*/

body {

font-family: Verdana, sans-serif;

margin: 0;

padding: 0;

height: 100vh;

overflow: hidden;

display: flex;

justify-content: center;

align-items: center;

flex-direction: column;

background-color: #000;

}

/\* Fona video \*/

video#background-video {

position: fixed;

top: 0;

left: 0;

width: 100%;

height: 100%;

object-fit: cover;

z-index: -1;

}

/\* Virsraksts \*/

h1 {

font-size: clamp(1rem, 5vmin, 3rem);

font-family: sans-serif;

color: #fff;

text-align: center;

margin-bottom: 40px;

}

.underlined {

position: relative;

}

.underline-a:after {

content: '';

position: absolute;

top: 120%;

width: 180%;

aspect-ratio: 3 / 1;

left: 50%;

transform: translate(-50%, 0);

border-radius: 50%;

border: 6px solid hsl(202, 82%, 56%);

clip-path: polygon(0 0, 50% 50%, 100% 0);

}

/\* Lodziņš saturam \*/

.box {

width: 30rem;

height: 20rem;

background-color: rgba(255, 255, 255, 0.2);

color: #fff;

border: 2px solid #fff;

padding: 1rem;

box-sizing: border-box;

text-align: center;

overflow-y: auto;

}

/\* Meklēšanas josla \*/

.search-bar {

width: calc(100% - 2rem);

padding: 0.5rem;

margin-bottom: 1rem;

border: 1px solid #fff;

border-radius: 4px;

background-color: rgba(255, 255, 255, 0.2);

color: #fff;

}

.search-bar::placeholder {

color: #fff;

}

/\* Elementu saraksts \*/

.elements {

margin-top: 1rem;

text-align: left;

}

/\* Elementu vienības \*/

.element-item {

display: flex;

justify-content: space-between;

align-items: center;

padding: 0.5rem 0;

border-bottom: 1px solid rgba(255, 255, 255, 0.5);

color: #fff;

}

.element-item:last-child {

border-bottom: none;

}

/\* Datuma stils \*/

.date {

font-size: 0.9rem;

color: rgba(255, 255, 255, 0.7);

}

/\* Atpakaļ pogas stils \*/

.back-button {

top: 1rem;

left: 1rem;

position: absolute;

padding: 0.5rem 1rem;

font-size: 18px;

background-color: #ff2f2f52;

color: #fff;

border: none;

border-radius: 4px;

cursor: pointer;

z-index: 10;

}

.button:hover {

background-color: #ff5050;

}

1. pielikums **combine\_elements.php**

<?php

session\_start();

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

include 'db.php'; // Datubāzes savienojums

if ($\_SERVER['REQUEST\_METHOD'] === 'POST') {

$element1 = $\_POST['element1'] ?? null;

$element2 = $\_POST['element2'] ?? null;

// Pārbauda, vai lietotājs ir pieteicies

if (!isset($\_SESSION['user\_id'])) {

echo json\_encode(["status" => "error", "message" => "User not logged in"]);

exit();

}

$userId = $\_SESSION['user\_id']; // Lietotāja ID

if ($element1 && $element2) {

// OpenAI API izmantošana elementu apvienošanai

$apiKey = "sk-proj-ardI4XzfcVJkmiWHN5fArpWDBHsWOgHAqPYXG7BHnRYOkxgQA2\_Wx0EcGBbcb\_84gWlTtgRaW0T3BlbkFJ4MGclayHx1zsvugqkgfqzlJ\_5WAfuc4UHWvEBAVahVLTxxC2iUYHIKkQ1lu\_1CvIZJsGqK5ssA";

$prompt = "Tu esi spēles amatniecības (crafting) sistēma. Tavs uzdevums ir apvienot divus elementus, ko nodrošina spēlētājs, un radīt jaunu, loģisku rezultātu. Ņem vērā šādus noteikumus:

1. Katrai kombinācijai ir jāseko loģiskai, dabiskai vai radošai secībai.

2. Izmanto emocijzīmes (emojis), lai attēlotu iegūto elementu, ja tas ir piemēroti, un neizmanto vairāk kā 2 emocijzīmes.

3. Atbildē sniedz tikai emocijzīmi un rezultātu, bez pieturzīmēm.

Piemēri:

- Apvienot uguni un ūdeni: 🌫️ tvaiks

- Apvienot zemi un uguni: 🌋 lava

- Apvienot gaisu un zemi: 🌪️ tornādo

Tagad apvieno šādus elementus: \"$element1\" un \"$element2\"";

// API pieprasījums

$ch = curl\_init();

curl\_setopt($ch, CURLOPT\_URL, "https://api.openai.com/v1/chat/completions");

curl\_setopt($ch, CURLOPT\_RETURNTRANSFER, 1);

curl\_setopt($ch, CURLOPT\_POST, 1);

curl\_setopt($ch, CURLOPT\_HTTPHEADER, [

"Content-Type: application/json",

"Authorization: Bearer $apiKey"

]);

curl\_setopt($ch, CURLOPT\_POSTFIELDS, json\_encode([

"model" => "gpt-4",

"messages" => [

["role" => "system", "content" => "You are a creative assistant."],

["role" => "user", "content" => $prompt]

],

"max\_tokens" => 50,

"temperature" => 0.7

]));

$response = curl\_exec($ch);

if (curl\_errno($ch)) {

echo json\_encode(["status" => "error", "message" => curl\_error($ch)]);

curl\_close($ch);

exit;

}

curl\_close($ch);

$response = json\_decode($response, true);

if (isset($response['choices'][0]['message']['content'])) {

$newElement = trim($response['choices'][0]['message']['content']);

// Pārbauda, vai jaunais elements jau pastāv

$stmt = $conn->prepare("SELECT COUNT(\*) FROM elements WHERE name = ? AND users\_id = ?");

$stmt->bind\_param("si", $newElement, $userId);

$stmt->execute();

$stmt->bind\_result($count);

$stmt->fetch();

$stmt->close();

if ($count > 0) {

// Elements jau pastāv

echo json\_encode(["status" => "exists", "newElement" => $newElement]);

} else {

// Jauns elements tiek pievienots datubāzei

$stmt = $conn->prepare("INSERT INTO elements (name, users\_id) VALUES (?, ?)");

$stmt->bind\_param("si", $newElement, $userId);

if ($stmt->execute()) {

echo json\_encode(["status" => "success", "newElement" => $newElement]);

} else {

echo json\_encode(["status" => "error", "message" => $stmt->error]);

}

$stmt->close();

}

} else {

echo json\_encode(["status" => "error", "message" => "OpenAI API failed."]);

}

} else {

echo json\_encode(["status" => "error", "message" => "Invalid input"]);

}

} else {

echo json\_encode(["status" => "error", "message" => "Invalid request method"]);

}

$conn->close();

?>

1. pielikums **db.php**

<?php

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

$servername = "localhost";

$username = "root";

$password = "";

$dbname = "b.iespejas";

$conn = new mysqli($servername, $username, $password, $dbname);

// Parbauda savienojumu

if ($conn->connect\_error) {

    die("Connection failed: " . $conn->connect\_error);

}

?>

1. pielikums **fetch\_elements.php**

<?php

session\_start();

include 'db.php';

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

if (!isset($\_SESSION['user\_id'])) {

echo json\_encode(["status" => "error", "message" => "User not logged in"]);

exit();

}

$userId = $\_SESSION['user\_id'];

$sql = "SELECT name FROM elements WHERE users\_id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("i", $userId);

$stmt->execute();

$result = $stmt->get\_result();

$userElements = [];

while ($row = $result->fetch\_assoc()) {

$userElements[] = $row;

}

echo json\_encode($userElements);

$conn->close();

?>

1. pielikums **fetch\_library\_elements.php**

<?php

session\_start();

header("Content-Type: application/json");

include 'db.php';

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

if (!isset($\_SESSION['user\_id'])) {

echo json\_encode(["status" => "error", "message" => "User not logged in"]);

exit();

}

$userId = $\_SESSION['user\_id'];

$sql = "SELECT name FROM saved\_elements WHERE users\_id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("i", $userId);

$stmt->execute();

$result = $stmt->get\_result();

$elements = [];

while ($row = $result->fetch\_assoc()) {

$elements[] = $row;

}

echo json\_encode($elements);

$conn->close();

?>

1. pielikums **fetch\_saved\_elements.php**

<?php

session\_start();

header("Content-Type: application/json");

include 'db.php';

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

if (!isset($\_SESSION['user\_id'])) {

echo json\_encode(["status" => "error", "message" => "User not logged in"]);

exit();

}

$userId = $\_SESSION['user\_id'];

$sql = "SELECT name FROM elements WHERE users\_id = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("i", $userId);

$stmt->execute();

$result = $stmt->get\_result();

$userElements = [];

while ($row = $result->fetch\_assoc()) {

$userElements[] = $row; // Each row is like ["name" => "💧 ūdens"]

}

echo json\_encode($userElements);

$conn->close();

1. pielikums **fetch\_saved\_game.php**

<?php

session\_start();

include 'db.php';

$response = ["hasSavedGame" => false];

if (isset($\_SESSION['user\_id'])) {

$userId = $\_SESSION['user\_id'];

$query = "SELECT \* FROM saved\_games WHERE user\_id = ? LIMIT 1";

$stmt = $conn->prepare($query);

$stmt->bind\_param("i", $userId);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows > 0) {

$response["hasSavedGame"] = true;

}

}

echo json\_encode($response);

?>

1. pielikums **game.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Spēles Logs</title>

<link rel="stylesheet" href="styles/game.css"> <!-- Stilu fails -->

</head>

<body>

<!-- Fona video -->

<video id="background-video" autoplay loop muted>

<source src="media/background.mp4" type="video/mp4">

</video>

<!-- Atpakaļ poga -->

<button id="return-button" class="button" onclick="goBack()">X</button>

<!-- Spēles konteiners -->

<div class="game-container" id="game-container">

<!-- Dzēšanas laukums -->

<div class="clear-area" id="clear-area">

Notīrīt šeit

</div>

<!-- Poga "Notīrīt visu" -->

<button class="clear-all-button" id="clear-all-button">

<img src="media/clear-icon.png" alt="Clear All" class="button-image">

</button>

<!-- Saglabāšanas laukums -->

<div class="save-area" id="save-area">

Saglabāt šeit

</div>

<!-- Elementu saraksts -->

<div class="element-list" id="element-list">

<input type="text" class="search-bar" placeholder="Meklēt..."> <!-- Meklēšanas josla -->

<!-- Elementi tiks ielādēti dinamiski -->

</div>

</div>

<script src="scripts/game.js"></script> <!-- JavaScript fails -->

</body>

</html>

1. pielikums **homepage.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Sākumlapa</title>

<link rel="stylesheet" href="styles/homepage.css">

</head>

<body>

<video id="background-video" autoplay loop muted>

<source src="media/background.mp4">

</video>

<div class="homepage">

<h1><span class="underlined underline-a">B.Iespējas</span></h1>

<button onclick="navigateToGame()">Spēle</button>

<button onclick="navigateToLibrary()">Bibliotēka</button>

<button onclick="logout()">Izrakstīties</button>

</div>

<script src="scripts/homepage.js"></script>

</body>

</html>

1. pielikums **index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Reģistrācija un Pieteikšanās</title>

<link rel="stylesheet" href="styles/index.css">

</head>

<body>

<video id="background-video" autoplay loop muted>

<source src="media/background.mp4">

</video>

<!-- Reģistrācijas logs -->

<div id="registration" class="container">

<h1><span class="underlined underline-a">Reģistrēties</span></h1>

<form action="register.php" method="POST">

<div class="form-group">

<label for="email">E-pasts</label>

<input type="email" id="email" name="email" placeholder="Ievadiet savu e-pastu" required>

</div>

<div class="form-group">

<label for="password">Parole</label>

<input type="password" id="password" name="password" placeholder="Izveidojiet paroli" required>

<div id="password-requirements">

<span id="length" class="requirement invalid">8+ rakstzīmes</span>

<span id="uppercase" class="requirement invalid">1 lielais burts</span>

</div>

</div>

<div class="form-group">

<label for="confirm-password">Atkārtot paroli</label>

<input type="password" id="confirm-password" name="confirm-password" placeholder="Atkārtojiet paroli" required>

</div>

<button type="submit">Reģistrēties</button>

</form>

<div class="text">

Jau esi reģistrējies? <a href="#" onclick="toggleForm('login')">Pieteikties</a>

</div>

</div>

<!-- Pieteikšanās logs -->

<div id="login" class="container" style="display: none;">

<h1><span class="underlined underline-a">Pieteikties</span></h1>

<form action="login.php" method="POST">

<div class="form-group">

<label for="login-email">E-pasts</label>

<input type="email" id="login-email" name="email" placeholder="Ievadiet savu e-pastu" required>

</div>

<div class="form-group">

<label for="login-password">Parole</label>

<input type="password" id="login-password" name="password" placeholder="Ievadiet paroli" required>

</div>

<button type="submit">Pieteikties</button>

</form>

<div class="text">

Nav izveidots profils? <a href="#" onclick="toggleForm('registration')">Reģistrēties</a>

</div>

</div>

<script src="scripts/index.js"></script>

</body>

</html>

1. pielikums **library.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Bibliotēka</title>

<link rel="stylesheet" href="styles/library.css">

</head>

<body>

<video id="background-video" autoplay loop muted>

<source src="media/background.mp4" type="video/mp4">

</video>

<button class="button back-button" onclick="goBack()">X</button>

<h1><span class="underlined underline-a">Bibliotēka</span></h1>

<div class="box">

<input type="text" class="search-bar" placeholder="Meklēt...">

<div class="elements"><!-- Saglabātie elementi tiks ielādēti šeit --></div>

</div>

<script src="scripts/library.js"></script>

</body>

</html>

1. pielikums **login.php**

<?php

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

session\_start(); // Start the session

include 'db.php';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

// Retrieve user inputs

$email = $\_POST['email'];

$password = $\_POST['password'];

// Fetch the user from the database

$sql = "SELECT \* FROM users WHERE email = ?";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("s", $email);

$stmt->execute();

$result = $stmt->get\_result();

if ($result->num\_rows === 1) {

$user = $result->fetch\_assoc();

// Verify the entered password against the hashed password

if (password\_verify($password, $user['password'])) {

// Save the user ID in the session

$\_SESSION['user\_id'] = $user['id'];

// Redirect to the homepage on success

header("Location: homepage.html");

exit;

} else {

echo "Invalid password! <a href='index.html'>Try again</a>";

}

} else {

echo "User not found! <a href='index.html'>Register here</a>";

}

// Close the statement and connection

$stmt->close();

$conn->close();

}

?>

1. pielikums **logout.php**

<?php

session\_start(); // Uzsāk sesiju

// Iznīcina sesiju un visus sesijas datus

session\_unset();

session\_destroy();

// Pāradresē uz pieteikšanās lapu

header("Location: index.html");

exit();

?>

1. pielikums **register.php**

<?php

session\_start(); // ✅ Required before setting session vars

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

include 'db.php';

if ($\_SERVER["REQUEST\_METHOD"] == "POST") {

$email = $\_POST['email'];

$password = $\_POST['password'];

$confirmPassword = $\_POST['confirm-password'];

if ($password !== $confirmPassword) {

echo "Paroles nesakrīt! <a href='index.html'>Mēģiniet vēlreiz</a>";

exit;

}

$hashedPassword = password\_hash($password, PASSWORD\_DEFAULT);

$sql = "INSERT INTO users (email, password) VALUES (?, ?)";

$stmt = $conn->prepare($sql);

$stmt->bind\_param("ss", $email, $hashedPassword);

if ($stmt->execute()) {

$userId = $stmt->insert\_id;

$basicElements = ['🔥 uguns', '💧 ūdens', '🌎 zeme', '💨 gaiss'];

foreach ($basicElements as $element) {

$check = $conn->prepare("SELECT 1 FROM elements WHERE name = ? AND users\_id = ?");

$check->bind\_param("si", $element, $userId);

$check->execute();

$check->store\_result();

if ($check->num\_rows === 0) {

$insert = $conn->prepare("INSERT INTO elements (name, users\_id) VALUES (?, ?)");

$insert->bind\_param("si", $element, $userId);

$insert->execute();

$insert->close();

}

$check->close();

}

// ✅ Automatically log in and go to game

$\_SESSION['user\_id'] = $userId;

$\_SESSION['email'] = $email;

header("Location: game.html");

exit;

} else {

echo "Kļūda: " . $stmt->error . "<br><a href='index.html'>Mēģiniet vēlreiz</a>";

}

$stmt->close();

$conn->close();

}

?>

1. pielikums **save\_element.php**

<?php

session\_start();

header("Content-Type: application/json");

include 'db.php';

error\_reporting(E\_ALL);

ini\_set('display\_errors', 1);

if (!isset($\_SESSION['user\_id'])) {

echo json\_encode(["status" => "error", "message" => "User not logged in"]);

exit();

}

if (!isset($\_POST['name'])) {

echo json\_encode(["status" => "error", "message" => "No element name provided"]);

exit();

}

$userId = $\_SESSION['user\_id'];

$name = $\_POST['name'];

// Check if already saved

$check = $conn->prepare("SELECT 1 FROM saved\_elements WHERE name = ? AND users\_id = ?");

$check->bind\_param("si", $name, $userId);

$check->execute();

$check->store\_result();

if ($check->num\_rows > 0) {

echo json\_encode(["status" => "exists"]);

$check->close();

exit();

}

$check->close();

// Save element

$save = $conn->prepare("INSERT INTO saved\_elements (name, users\_id) VALUES (?, ?)");

$save->bind\_param("si", $name, $userId);

if ($save->execute()) {

echo json\_encode(["status" => "success"]);

} else {

echo json\_encode(["status" => "error", "message" => "Failed to save element"]);

}

$save->close();

$conn->close();

?>