****

Liepājas Valsts tehnikums

**Aviācijas vārdnīcas mājaslapa un tās pārvaldības sistēma**

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija Programmēšanas tehniķis

Grupas nosaukums 4PT

|  |  |
| --- | --- |
| Projekta izstrādātājs | ………………………………………………  /vārds, uzvārds, paraksts/ |

## Aizstāvēšanas datums 2023. gada

Liepāja 2023

# Saturs

[Ievads 3](#_Toc130815817)

[1. Uzdevuma formulējums 4](#_Toc130815818)

[2. Programmatūras prasību specifikācija 5](#_Toc130815819)

[2.1. Produkta perspektīva 5](#_Toc130815820)

[2.2. Funkcionālās prasības 5](#_Toc130815821)

[2.3. Pārvaldības sistēmas funkcionālās prasības 6](#_Toc130815829)

[2.4. Nefunkcionālās prasības 9](#_Toc130815838)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 10](#_Toc130815839)

[4. Sistēmas modelēšana un projektēšana 12](#_Toc130815840)

[5. Lietotāju ceļvedis 13](#_Toc130815841)

[6. Testēšanas dokumentācija 14](#_Toc130815842)

[7. Secinājumi 15](#_Toc130815843)

[8. Lietoto saīsinājumu saraksts 16](#_Toc130815844)

[9. Izmantotā literatūra un informācijas avoti 17](#_Toc130815845)

[Pielikumi 18](#_Toc130815846)

# Ievads

Abreviatūras palīdz vienkāršot valodu, palielināt efektivitāti un ietaupīt laiku, izmantojot garāku vārdu vai frāžu īsākas versijas. Tomēr, ja tiek izmantots tik daudz saīsinājumu, var būt grūti izsekot tiem visiem. Saīsinājumu vārdnīca ir vērtīgs uzziņu rīks, kurā ir uzskaitīti visbiežāk lietotie saīsinājumi, kā arī to nozīme un definīcijas.

Projekts “Abbreviation dictionary” ir web balstīts projekts kurā ieviesta mājaslapa un tā satur abreviatūru vārdnīcu par aviāciju. Projekta galvenais mērķis ir piesaistīt konkrētu auditoriju kuri vēlas apgūt abreviatūras un pārbaudīt savas zināšanas ar testa palīdzību.

Mājaslapa kalpo kā vārdnīca un tests, kuru jebkurš var piekļūt un pārbaudīt savas zināšanas par apgūtajām abreviatūrām testa veidā kurā izvēlas vienu no pareizajām atbildēm.

Mājaslapā ir pieejama pārvaldības sistēma kura ir pieejama administratoram kurā var pievienot jaunus terminus, tos dzēst vai rediģēt, kā arī pievienot jaunas kategorijas, lai nākotnē būtu iespēja papildināt vārdnīcu ar citu specialitāšu terminiem un abreviatūrām.

# Uzdevuma formulējums

Liepājas Valsts tehnikuma direktora vietniece uzdeva man izveidot vārdnīcu par aviāciju, (nākotnē varētu pievienot arī citas specialitātes terminus) kurus pašlaik varēs apgūt audzēkņi kuri vēlas apgūt aviācijas zināšanas. Mans uzdevums ir izveidot mājaslapu par terminiem kura var veikt nepieciešamās funkcijas, lai mājaslapas lietotāji varētu apgūt terminus un pārbaudīt savas zināšanas. Lietotājiem mājaslapa sastāv no divām galvenajām lapām:

* Terminu lapa
* Testu lapa

Mājaslapa jāveido tā, lai uzreiz būtu skaidrs kā, kur, un ko atrast. Terminu lapa būs veidota tā, lai tajā varētu ērti izvēlēties terminus pa kategorijām, lai tos vieglāk apskatīt, daļai no terminiem ir pieejama vairāk informācija ja uzspiež uz pogu more info, lielai daļai no terminiem nebija iespēja pievienot papildus informāciju, jo daudzi termini ir sarežģītāki un par tiem nav daudz informācijas ko varētu atrast interneta resursos, kā piemēram vikipēdija.

Testu lapā būs iespēja pārbaudīt savas zināšanas par nejaušiem terminiem.

Būs pieejama arī pārvaldības sistēma kas pieejama tikai virslietotājiem jeb administratoram, ar kuras palīdzību var pievienot jaunas kategorijas, lai nākotnē būtu iespēja papildināt vārdnīcu ar citu specialitāšu terminiem un abreviatūrām, kā arī pievienot vai rediģēt jaunus terminus.

# Programmatūras prasību specifikācija

Šajā nodaļā būs apraksts par projektu “Abbreviation dictionary” kurā galvenie lietotāji būs gaisa kuģa mehāniķi.

## Produkta perspektīva

Mājaslapas galvenā funkcija ir piesaistīt lietotājus kuri vēlas apgūt terminus. Mājaslapa nodrošina visus vajadzīgos terminus kuri nepieciešami par konkrēto kategoriju. Projekts “Abbreviation dictionary” ir nepieciešams, jo šāds projekts skolai vēl nav izveidots un tas ir nepieciešams, lai varētu apgūt terminus.

## Funkcionālās prasības

### Kategorijas izvēle

Mērķis: Lietotājs atver vienu no kategorijām.

Ievaddati: Klikšķis uz kādu no kategorijām

Apstrāde: Lietotājs tiek novirzīts uz izvēlēto kategoriju.

Izvaddati: Atveras izvēlētā kategorija

### Konkrēta termina meklēšana

Mērķis: Meklēt terminu.

Ievaddati: Klikšķis uz “Search data” pogas un ieraksta kuru terminu meklēt, klikšķis uz “Search” pogas

Apstrāde: Izpildās funkcija kura sameklē un izvada visus terminus kurus meklē.

Izvaddati: Parādās termins/i kurus meklē

### Kategorijas burta izvēle

Mērķis: Lietotājs atver vienu no burtiem.

Ievaddati: Klikšķis uz kādu no burtiem

Apstrāde: Lietotājs tiek novirzīts uz izvēlēto burta kategoriju.

Izvaddati: Atveras izvēlētā burta abreviatūra

### Vairāk informācijas

Mērķis: Lietotājs atver Online source/more info.

Ievaddati: Klikšķis uz more info

Apstrāde: Lietotājs tiek novirzīts uz online source saiti.

Izvaddati: Atveras saite uz izvēlētās abreviatūras vikipēdijas vai citas informatīvas saites

### Burta izvēle

Mērķis: Lietotājs atgriežas pie burta izvēles.

Ievaddati: Klikšķis uz back pogu

Apstrāde: Lietotājs tiek nosūtīts atpakaļ uz kategorijas izvēli.

Izvaddati: Atveras iepriekš izvēlētās kategorijas burtu izvēle

### Sākuma lapa

Mērķis: Lietotājs atgriežas sākuma lapā.

Ievaddati: Klikšķis uz home pogu

Apstrāde: Lietotājs tiek nosūtīts atpakaļ uz sākuma lapu.

Izvaddati: Atveras sākuma lapa

### Tests

Mērķis: Atvērt testa lapu.

Ievaddati: Klikšķis uz test pogu

Apstrāde: Lietotājs tiek nosūtīts uz testa lapu.

Izvaddati: Atveras testa lapa

### Testa pildīšana

Mērķis: Pildīt testu.

Ievaddati: klikšķis uz kādu no atbildēm uz katru terminu kurš ir testā, pēc pabeigšanas klikšķis uz submit.

Apstrāde: Tiek pārbaudītas lietotāja atbildes un izvada rezultātu.

Izvaddati: tiek parādīts rezultāts par testu

## Pārvaldības sistēmas funkcionālās prasības

### Piekļuve sistēmai

Mērķis: Apmeklēt mājaslapas pārvaldības sistēmu.

Ievaddati: Klikšķis uz login pogu

Apstrāde: Lietotājs tiek nosūtīts uz login lapu.

Izvaddati: Atveras login lapa

### Autorizēšanās sistēmā

Mērķis: Pieslēgties sistēmā.

Ievaddati: Ievada Lietotājvārdu un Paroli tam paredzētā vietā

Apstrāde: Tiek pārbaudīts Lietotājvārds un parole, ja ievaddati sakrīt ar datu bāzi tad notiek autorizēšanās.

Izvaddati: Atveras sākuma lapa

### Kategorijas pievienošana

Mērķis: Pievienot jaunu kategoriju.

Ievaddati: Ieraksta kategorijas nosaukumu tam paredzētā vietā un klikšķis uz submit

Apstrāde: Tiek veikta pārbaude vai lietotājs nav atstājis tukšu lauku pirms submit pogas nospiešanas, attiecīgi, izpildās funkcija.

Izvaddati: Tiek pievienota jauna kategorija

### Termina pievienošana

Mērķis: Pievienot jaunu terminu.

Ievaddati: Ieraksta saīsinājumu, terminu, kategoriju, kategorijas burtu un online source tam paredzētā vietā un klikšķis uz submit

Apstrāde: Tiek veikta pārbaude vai visi obligātie lauki ir aizpildīti, izpildās konkrētā funkcija.

Izvaddati: tiek pievienots jauns termins

### Termina dzēšana

Mērķis: Izdzēst terminu kurš nav nepieciešams.

Ievaddati: Ieraksta saīsinājumu, terminu, kategoriju kuru vēlas izdzēst un klikšķis uz delete pogu.

Apstrāde: Tiek pārbaudīts vai ir aizpildīti visi obligātie lauki, izpildās konkrētā funkcija.

Izvaddati: Tiek izdzēsts termins

### Termina meklēšana, lai to rediģētu

Mērķis: meklē terminu.

Ievaddati: Ieraksta termia saīsinājumu kuru vēlas izlabot.

Apstrāde: Izpildās konkrētā funkcija kura sameklē terminu.

Izvaddati: Atveras termina rediģēšanas lapa.

### Termina rediģēšana

Mērķis: rediģēt terminu.

Ievaddati: Veic labojumus.

Apstrāde: Tiek pārbaudīts vai ir aizpildīti visi obligātie lauki, izpildās konkrētā funkcija kura saglabā izmaiņas.

Izvaddati: Tiek rediģēts termins

## Nefunkcionālās prasības

* Veiktspēja: vietne ir jāielādē noteiktā laika limitā (piemēram, 2 sekundes), lai nodrošinātu ātru un atsaucīgu lietotāja pieredzi.
* Pieejamība: vietnei jābūt pieejamai visu diennakti bez dīkstāves, izņemot plānotos apkopes periodus.
* Mērogojamība: vietnei jāspēj apkalpot arvien lielāku lietotāju skaitu un trafiku, nepasliktinot veiktspēju.
* Lietojamība: tīmekļa vietnei ir jābūt viegli orientējamai un lietojamai, ar skaidru un kodolīgu valodu, un tai jābūt pieejamai lietotājiem ar invaliditāti.
* Saderība: vietnei jābūt saderīgai ar dažādām pārlūkprogrammām, operētājsistēmām un ierīcēm, tostarp galddatoriem un mobilajām ierīcēm.
* Uzticamība: vietnei jābūt stabilai un uzticamai, ar zemu kļūdu un avāriju līmeni, un tai jābūt viegli atkopjamai kļūmes gadījumā.
* Uzturamība: vietnei jābūt veidotai tā, lai tā būtu vienkārša apkope un atjaunināšana, ar skaidru un kodolīgu dokumentāciju un ērti lietojamiem administrēšanas rīkiem.

# Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

Uzsākot projekta plānošanas gaitu un meklējot tematam piemērotāko projekta risinājumu, atradu lielu klātu ar iespējamajiem risinājumiem, kas bija piemēroti gan mājaslapas saskarnes daļai, gan arī, lai veidotu mājaslapas funkcionālo daļu. Tādēļ es priekš sevis nostādiju vadlīnijas, pēc kurām es izvēlējos atbilstošākos risinājuma variantus. Šīs vadlīnijas ietver sekojošos nosacījumus:

* Ieteicams, ka man ar šo līdzekli ir bijusi pieredze vai risinājums ar kuru jau iepriekš esmu strādājis, kā arī tās programmēšanas valodas ar kurām man ir pieredze.
* Līdzeklim jābūt ar apjomīgu un labu dokumentāciju vai arī ar plaši pieejamu informāciju interneta vietnēs, gadījumā, ja rodas neskaidrības.

## 3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

Projekta izveidei izmantoju šādus līdzekļus:

* Visual Studio Code
* MySQL Workbench

Pirmo izstrādes vide ir Visual Studio Code kurā es izstrādāju visu sistēmu, šo vidi es izvēlējos, jo iepriekš esmu šajā vidē strādājis un mācījies, līdz ar to, esmu pieradis pie programmas un visām tām piedāvātajām iespējām. Kā arī VS Code ir bezmaksas izstrādes vide ar daudz iespējām.

Otrā izstrādes vide ir MySQL Workbench kurā es izstrādāju datu bāzi ar visām tās funkcijām. Šo vidi izvēlējos, jo man šajā vidē ir pieredze, esmu iepriekš strādājis un izmantojis, tāpēc labi pārzinu šo vidi un varēju to izmantot.

Izmantotās programmēšanas valodas un līdzekļi:

* HTML – mājaslapas struktūrai.

(HTML ir standarta iezīmēšanas valoda, ko izmanto, lai izveidotu tīmekļa lapu struktūru un parādītu to saturu. Tas ir pamats lielākajai daļai tīmekļa lapu internetā. HTML ļauj definēt tīmekļa lapas struktūru, izmantojot tagus un elementus, kas nosaka, kā tīmekļa pārlūkprogrammā jāparāda dažādas satura daļas.)

* CSS – mājaslapas noformēšanai.

(CSS ir stila lapu valoda, ko izmanto, lai aprakstītu HTML dokumentu noformējumu un izskatu. Tā ir tīmekļa dizaina pamattehnoloģija un ļauj kontrolēt tīmekļa lapu vizuālos aspektus.)

* PHP – mājaslapas funkcionalitātei.

(PHP ir populāra servera puses skriptu valoda, ko galvenokārt izmanto tīmekļa izstrādei. To plaši izmanto dinamisku un interaktīvu tīmekļa lietojumprogrammu veidošanai)

* MySQL – datubāzes vaicājumu valoda.

(MySQL ir atvērtā koda relāciju datu bāzes pārvaldības sistēma, ko plaši izmanto strukturētu datu pārvaldībai un glabāšanai. Tā ir populāra tīmekļa lietojumprogrammu izvēle, un to bieži izmanto kopā ar PHP, lai izveidotu dinamiskas vietnes.)

## 3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts

Alternatīvas izstrādes vides kuras es varēju izmantot VS Code vietā:

* Notepad++
* Visual Studio

Gan Visual Studio Code un Notepad++ ir populāri teksta redaktori, taču ir vairāki iemesli, kāpēc es izvēlējos Visual Studio Code:

* VS Code piedāvā spēcīgu paplašinājumu ekosistēmu, kas uzlabo tā funkcionalitāti. Šie paplašinājumi aptver plašu vajadzību klāstu, tostarp valodas atbalstu, koda formatēšanu, atkļūdošanu, versiju kontroles integrāciju un daudz ko citu. Tā kā ir pieejami tūkstošiem paplašinājumu, izstrādātāji var pielāgot savu rediģēšanas pieredzi savām īpašajām prasībām.
* Integrētās izstrādes vides (IDE) līdzekļi: atšķirībā no Notepad++, VS Code nodrošina plašāku funkciju kopumu, kas padara to tuvāku pilnvērtīgam IDE. Tā piedāvā tādas funkcijas kā atkļūdošanas rīkus, integrētu termināli, versiju kontroles integrāciju (Git) un iebūvētu atbalstu populārām programmēšanas valodām. Šīs funkcijas uzlabo produktivitāti un racionalizē izstrādes darbplūsmu, īpaši sarežģītiem projektiem.
* Iebūvēta avota kontrole: VS Code ir iebūvēts atbalsts tādām populārām versiju kontroles sistēmām kā Git. Tas nodrošina īpašu avota vadības skatu, kas ļauj izstrādātājiem veikt kopīgas avota kontroles darbības, piemēram, veikt izmaiņas, pārskatīt atšķirības.

Lai gan Notepad++ joprojām ir viegls un efektīvs teksta redaktors, Visual Studio Code plašais funkciju komplekts, paplašināmība un vairāku platformu atbalsts padara to par vēlamo izvēli daudziem izstrādātājiem, īpaši tiem, kas strādā ar sarežģītiem projektiem vai kuriem nepieciešamas uzlabotas izstrādes iespējas.

Alternatīvas izstrādes vides kuras es varēju izmantot MySQL Workbench vietā:

* Microsoft SQL Server
* PHPMyAdmin

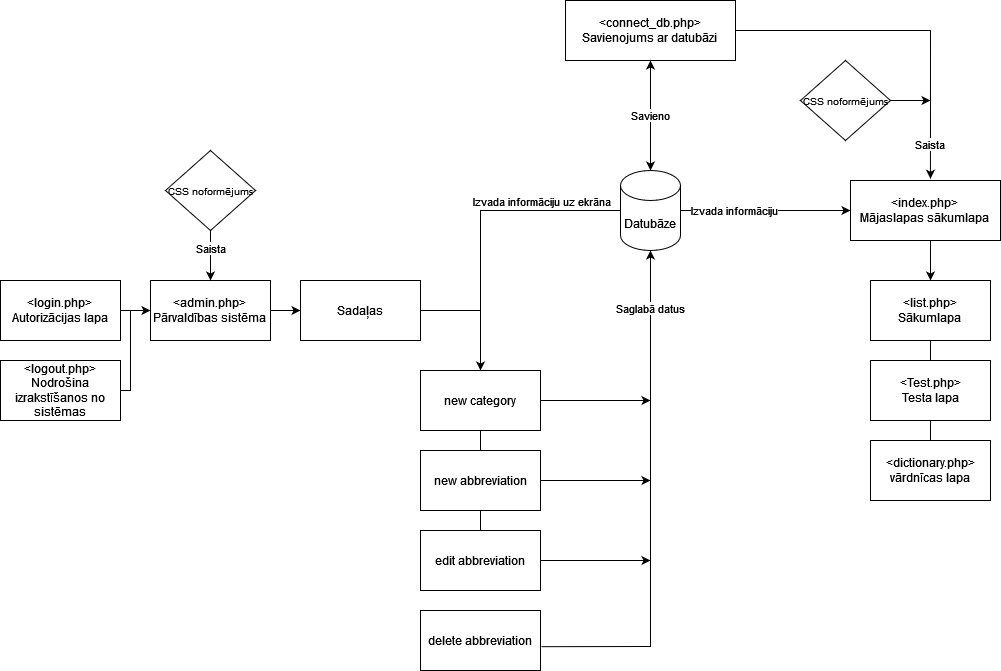
Kāpēc es izvēlējos MySQL:

MySQL ir relāciju datu bāzes pārvaldības sistēma, kas nodrošina uzticamu un mērogojamu risinājumu strukturētu datu pārvaldībai. Tas ir pazīstams ar savu veiktspēju, stabilitāti un lietošanas ērtumu. MySQL ir īpaši populārs tīmekļa izstrādē, pateicoties tā nevainojamai integrācijai ar skriptu valodām, piemēram, PHP, un tā spēj efektīvi apstrādāt lielu trafika apjomu. Lai gan MySQL galvenokārt ir vērsta uz datu bāzes pārvaldību, tā nenodrošina grafisko lietotāja interfeisu (GUI) administrēšanai.

# Sistēmas modelēšana un projektēšana

## Sistēmas struktūras modelis

Diagramma par sistēmas struktūras modeli:



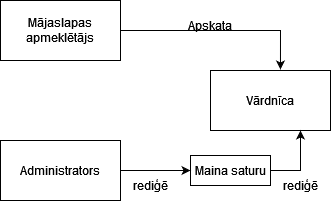
## ER diagramma

Diagramma par ERD:

## 

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

Diagramma par funkcionālo un dinamisko sistēmas modeli:

****

## Aktivitāšu diagramma

Aktivitāšu diagramma:

## 

## Lietojumgadījumu diagramma

Lietojumgadījumu diagramma:

## 

* 1. **Sistēmas moduļu apraksts un algoritmu shēmas**

Diagramma par sistēmas moduļu aprakstu un algoritmu shēmu:

## 

# Lietotāju ceļvedis

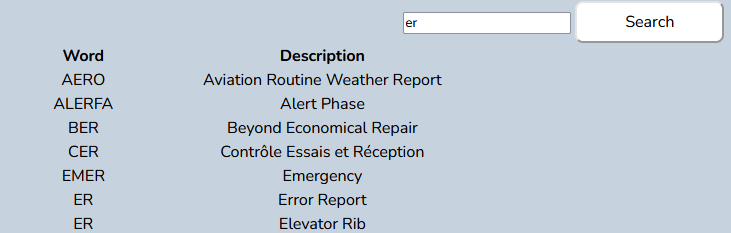
Izvēlas kategoriju:



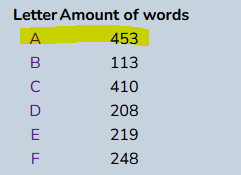
Meklē terminu:



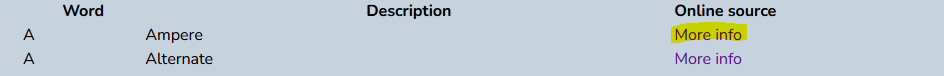
Termina meklēšana:



Izvēlas nākošo kategoriju:



Izvēlas vairāk informāciju:



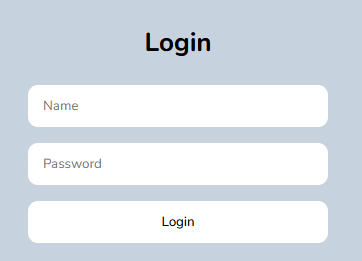
“Login”- admina ielogošanās, “Test” – testa daļa, “Back” – atgriešanās uz sākuma lapu:



Testa pildīšana:



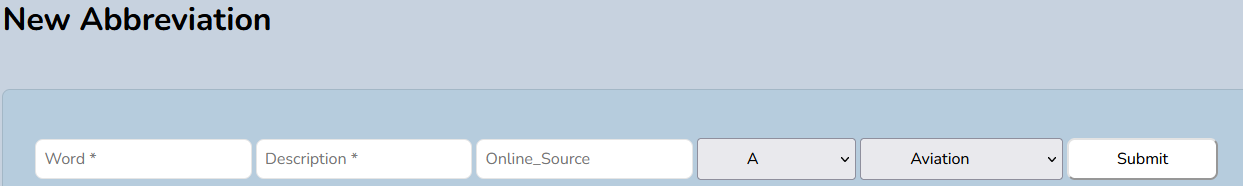
Admina ielogošanās:



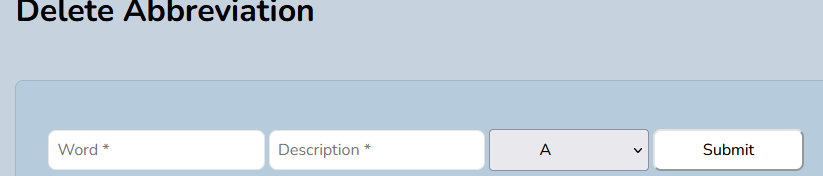
Pievieno jaunu kategoriju:



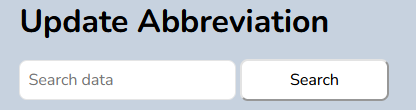
Pievieno jaunu terminu:



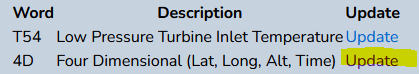
Izdzēš terminu:



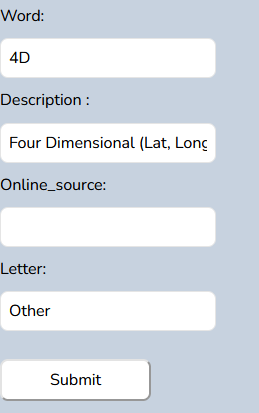
Meklē terminu kuru rediģēt:



Izvēlas terminu kuru rediģēt:



Termina rediģēšana:



Admina izlogošanās:



# Testēšanas dokumentācija

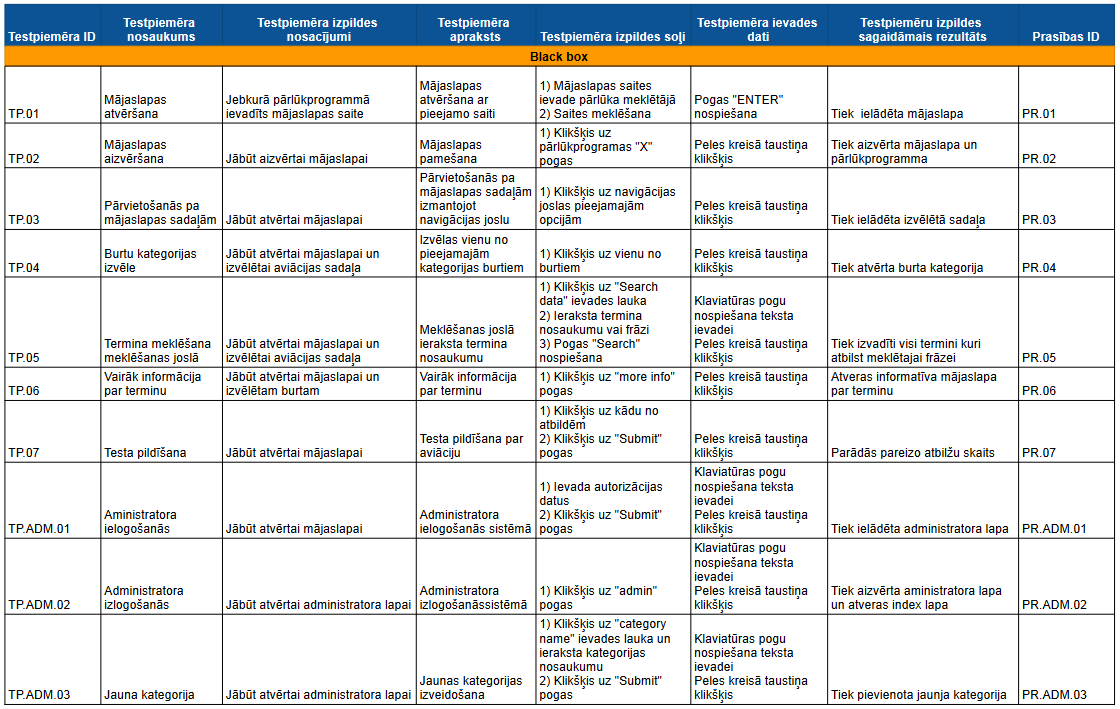
* 1. **Izvēlētās testēšanas metodes, rīku apraksts un pamatojums**

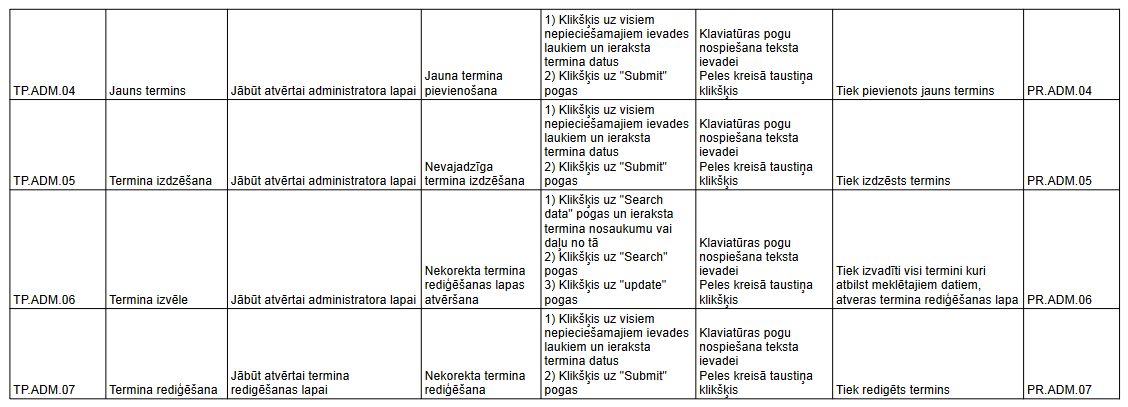
Abbreviation dictionary testēšanu veicu ar manuālo testēšanu, izmantoju melnās kastes testēšanas metodes, kuras testē galvenās funkcionālās prasības, kuru izpildes rezultātu salīdzinu ar gaidāmo rezultātu, un tādā veidā ir viegli saprast, vai ir sasniegts gaidāmais rezultāts.

Testpiemēru daļā izmantoto identifikātoru atšifrējumi:

* TP – testpiemērs
* ADM – administratora sadaļa
* PR – prasība

Testpiemēru kopa:





## Testēšanas žurnāls

## Testēšanas žurnālā parādīti testpiemēru kopas rezultāti:

## 

# Secinājumi

Veidojot vārdnīcu par aviāciju es sastapos ar dažādām problēmām kuras nācās atrisināt un iemācīeties, kā arī meklēt atrisinājumu internetā,

* Problēma ar termina rediģēšanu kuru sākumā bija jāraksta manuāli, proti, jāieraksta pats termins bez kļūdām un jaunais termina variants. Šo problēmu atrisināju ar termina meklēšanas joslu kurā ieraksta daļu no termina vai visu terminu un tad izvēlas to rediģēt, tādejādi to varēju izmantot arī vārdnīcas daļā kurā visi to funkciju var izmantot, lai vieglāk atrastu kādu konkrētu terminu.
* Problēma ar testa veidošanu kuru es vēlējos izveidot ar automātisku sistēmu, kura katru reizi kad to vēlas pildīt izvada dažādus jautājumus. Šo broblēmu neizdevās atrisināt, tāpēc testu izveidoju manuāli ar 10 dažādiem jautājumiem kuri katru reizi būs vieni un tie paši.
* Problēma ar jaunas kategorijas sadalīšana mazākās kategorijās (burtu kategorija), tā pat kā ar galveno aviācijas kategoriju. Jaunu kategoriju sadalīšana mazākās kategorijās neizdevās dēļ nekorektas datubāzes izveides, kuru es veidoju domājot ka būs tikai viena galvenā kategorija par aviāciju, bet veidojot šo projektu izdomāju, ka varētu pievienot citas kategorijas. Tāpēc kad izveido jaunu kategoriju, jaunie termini nebūs sadalīti un visi tiks izvadīti vienā lapā.

Mani sasniegumi bija tādi, ka apguvu vairāk prasmes un zināšanas par PHP valodu, kā arī izveidot vārdnīcu par aviāciju.

Mājaslapas dizainam nepievērsu daudz uzmanības un neko daudz nemainīju, jo galvenokārt strādāju pie mājaslapas funkcionalitātes.

# 8. Lietoto saīsinājumu saraksts

ADM - administratora sadaļa

CSS – Cascading Style Sheets (Stila lapas kaskadēšana)

HTML – hypertext markup language (Hiperteksta iezīmēšanas valoda)

PHP - Hypertext Preprocessor

PR - prasība

TP - testpiemērs

# 9. Izmantotā literatūra un informācijas avoti

Obligāto lauku aizpilde:

<https://www.w3docs.com/snippets/html/how-to-use-the-required-attribute-with-the-radio-input-field.html>

Testa veidošana:

<https://webdevtrick.com/simple-quiz-in-php-source-code/>

Datu uzlabošana:

<https://www.studentstutorial.com/php/php-mysql-data-update.php>

Meklēšanas josalas izveidošana:

<https://www.fundaofwebit.com/post/how-to-make-search-box-and-filter-data-in-html-table-from-database-in-php-mysql>

Meklēšanas josalas izveidošana:

<https://www.youtube.com/watch?v=yp5pYIg4WHc>

Uzglabāto procedūru izmantošana:

<https://www.invezzatechnologies.com/stored-procedure-lesson-in-php/>

Informatīva mājaslapa par HTML, CSS, PHP un MySQL:

<https://www.w3schools.com/>

Datu uzlabošana izmantojot PHP un MySQL:

<https://www.youtube.com/watch?v=J4nn9kxP1CE&t>

Savienošanās ar datubāzi:

<https://www.w3schools.com/php/php_mysql_connect.asp>

PHP globālie mainīgie:

<https://www.w3schools.com/php/php_superglobals.asp>

PHP globālie mainīgie:

<https://www.w3schools.com/php/php_superglobals_post.asp>

MySQL informācija:

<https://www.w3schools.com/mysql/mysql_delete.asp>

Informācija par CSS:

<https://www.w3schools.com/css/default.asp>

Sesiju sistēma:

<https://www.w3schools.com/php/php_sessions.asp>

Login sistēma ar sesiju palīdzību:

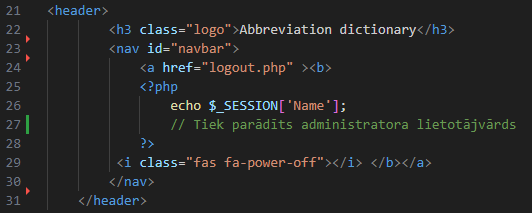
<https://phppot.com/php/php-login-script-with-session/>

Admin login, kā arī daži sīkumi ir ņemti no iepriekšējiem projektiem ko darīju skolā mācīšanās laikā.

# Pielikumi

**Komandrindas:**

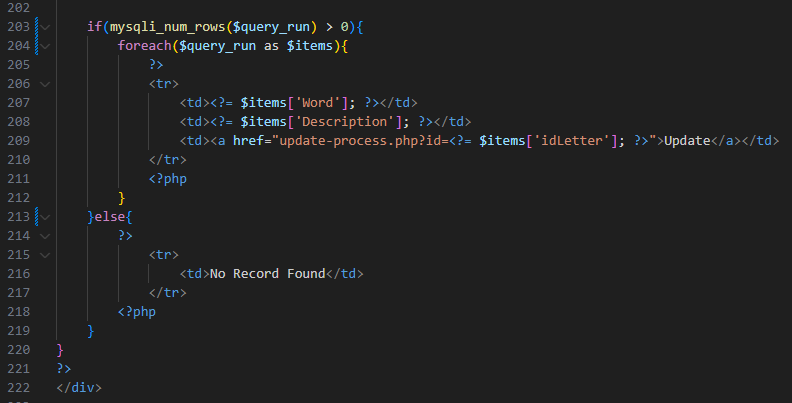
Administratora galvene:

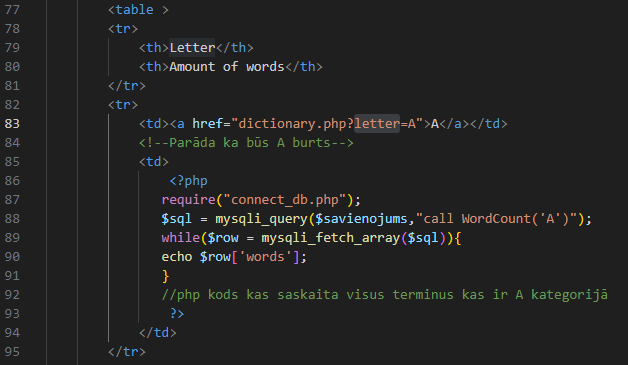


Kategorijas pievienošana:

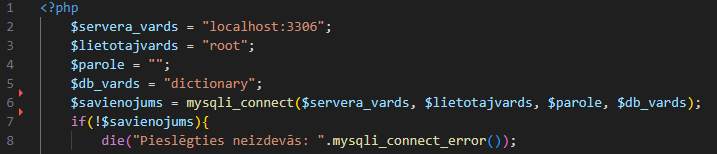


Termina meklēšana:

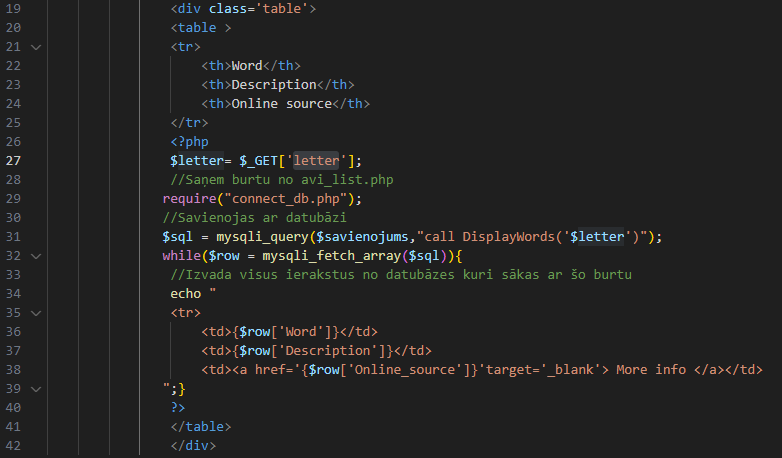


Burtu kategorijas izvadīšana: 

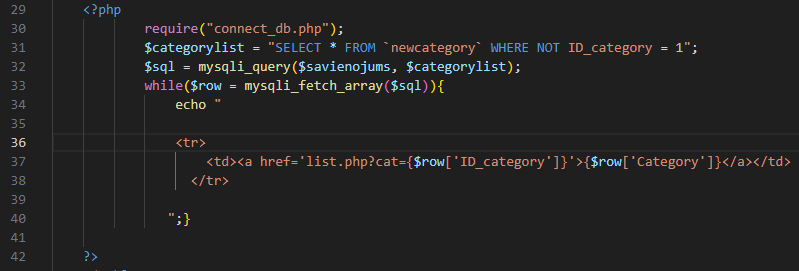
Savienošanās ar datu bāzi:



Izvada visus terminus:



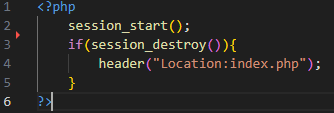
Izvada kategorijas:



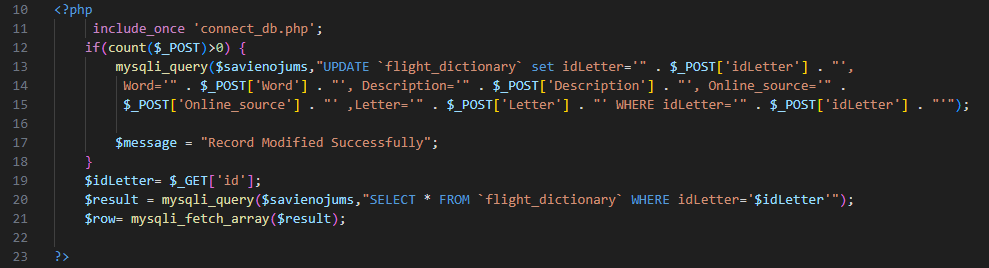
Administratora pieslēgšanās:



Admina izlogošanās:



Termina rediģēšana:





Mēģinājums pieslēgties administratora lapai bez piekļuves:

