Rīgas Tālmācības Vidusskola

Programmēšana II - 3

|  |  |
| --- | --- |
| Projekta izstrādātājs | Artūrs Javnošāns |

Liepāja 2023

# Saturs

[Saturs 2](#_Toc135866009)

[Ievads 4](#_Toc135866010)

[1. Uzdevuma formulējums 5](#_Toc135866011)

[2. Programmatūras prasību specifikācija 6](#_Toc135866012)

[2.1 Administrācija 6](#_Toc135866013)

[2.2 Lietotāja reģistrēšanās 6](#_Toc135866014)

[2.3 Lietotāja pieslēgšanās 7](#_Toc135866015)

[2.4 Problēma 7](#_Toc135866016)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 8](#_Toc135866017)

[3.1 Izvēlēto risinājumu līdzekļu un valodu apraksts 8](#_Toc135866018)

[3.2 Iespējamo (*alternatīvo)* risinājuma līdzekļu un valodu apraksts 10](#_Toc135866019)

[4. Sistēmas modelēšana un projektēšana 11](#_Toc135866020)

[4.1 Sistēmas struktūras modelis 11](#_Toc135866021)

[4.2 ER diagramma 12](#_Toc135866022)

[4.3 Funkcionālais un dinamiskais sistēmas modelis 15](#_Toc135866023)

[4.4 Aktivitāšu diagramma 16](#_Toc135866024)

[4.5 Lietotājgadijūma diagramma ( Use Case ) 18](#_Toc135866025)

[5. Sistēmas moduļu apraksts un algoritmu shēmas 19](#_Toc135866026)

[6. Lietotāju ceļvedis 19](#_Toc135866027)

[6.1 Pieslēgšanās 19](#_Toc135866028)

[6.2 Reģistrēšanās 21](#_Toc135866029)

[6.3 Navigācija 23](#_Toc135866030)

[6.4 Sākums 25](#_Toc135866031)

[6.5 Problēmas pievienošana 25](#_Toc135866032)

[6.6 Lietotāja profils 26](#_Toc135866033)

[6.7 Administrācijas panelis 29](#_Toc135866034)

[7. Testēšanas dokumentācija 31](#_Toc135866035)

[7.1 Melnās kastes testēšanas piemēri 31](#_Toc135866036)

[7.2 Navīgacijas jolsa ( Nepieslēdzies lietotājs ) 31](#_Toc135866037)

[7.3 Navigācijas josla ( Pieslēdzies lietotājs ) 32](#_Toc135866038)

[7.3 Lietotāja lapa 32](#_Toc135866039)

[8. Licences 34](#_Toc135866040)

[8.1 GNU Ģenerālā Publiskā Licence (GNU GPL) 34](#_Toc135866041)

[8.2 MIT Licence 34](#_Toc135866042)

[8.3 Salīdzinājums 34](#_Toc135866043)

[8.4 Piemērotākā licence 34](#_Toc135866044)

[9. Programmatūras kods 34](#_Toc135866045)

[9.1 Administratora ielogošanās 34](#_Toc135866046)

[9.2 Administrātora panelis 35](#_Toc135866047)

[9.3 Problēmas pievienošana. 38](#_Toc135866048)

[9.4 Ielogošanās profilā 40](#_Toc135866049)

[9.5 Lietotāja izlogošanās 41](#_Toc135866050)

[9.6 Reģistrēšanās 42](#_Toc135866051)

# Ievads

Šī mājaslapa nodrošina lietotājiem iespēju pieslēgties un reģistrēties, lai varētu dalīties savās IT problēmās un saņemt palīdzību no administrācijas. Lietotājiem ir pieejams lietotāja panelis, kur tie var pievienot savas IT problēmas un sagaidīt administrācijas atbildi.

Piesakoties mājaslapai, lietotājs var izveidot savu kontu, aizpildot reģistrācijas formu, kurā jānorāda nepieciešamie dati, piemēram, lietotājvārds, e-pasta adrese un parole. Pēc veiksmīgas reģistrācijas lietotājs var izmantot savas pieteiktās akreditācijas datus, lai pieslēgtos mājaslapai.

Pēc pieslēgšanās lietotājs iegūst pieeju savam lietotāja panelim, kurā var redzēt sarakstu ar savām iepriekš iesniegtajām IT problēmām un to statusu. Lietotājam ir iespēja pievienot jaunu problēmu, norādot tās raksturojumu un citu svarīgu informāciju. Kad problēma ir iesniegta, administrācija to pārskata un sniedz atbildi vai risinājumu.

Administrācijas pārstāvjiem ir atsevišķs, savādāks pieslēgšanās process. Viņi nevar reģistrēties paši, bet var izmantot tikai speciālu administrācijas pieslēgšanās formu. Pēc pieslēgšanās administrācija iegūst pieeju admina panelim, kurā var apskatīt visus iesniegtos IT problēmu ierakstus. Administrācija var rediģēt problēmu informāciju, mainīt to statusu un sniegt atbildes vai risinājumus problēmu iesniedzējiem.

Šī mājaslapa piedāvā lietotājiem iespēju dalīties ar savām IT problēmām, saņemt palīdzību no administrācijas un sekot līdzi problēmu atrisināšanas procesam. Administrācija var efektīvi pārvaldīt un atbildēt uz iesniegtajām problēmām, nodrošinot lietotājiem efektīvu un kvalitatīvu atbalstu.

1. Uzdevuma formulējums

Galējais uzdevums ir izstrādāt IT Atbalsta mājaslapu, kas atvieglo darbu uzņēmumam saņemt, apskatīt un veidot IT Atbalsta pasūtijumus. Savukārt, klientiem ir vieglāk pieteikt savu problēmu IT Adbalsta mājaslapā.

Izstrādāt mājaslapas dizainu un skicējumu.

1. Analīzēt un apgūt sistēmu pie kuras mājaslapa tiks veidota.
2. Modificēt un pielāgot to vietnes vajadzībām.
3. Izstrādāt mājaslapu balstoties pēc dizaina, funkcijām un skicējuma.
4. Izstrādāt un rediģet datubāzi mājaslapas vajadzībām un funkcijām.
5. Savienot izstrādāto datubāzi ar mājaslapu.
6. Izveidot lietotāja grupas ‘admin’ funkcijas un pieejas.
7. Izstrādāt un konfigurēt satura pārvaldības sistēmu, lai mājaslapu administratori varētu rediģēt statisko tekstu.
8. Veikt mājaslapas testēšanu un analīzi.
9. Novērst mājaslapas nepilnības un kļūdas.

Kad tiek izveidots pilnīgi darbspējīgs tīmekļa portāls ar visām iepriekš apspriestām funkcijām, mērķis projekta tiks sasniegts. Tīmekļa vietne būs pilnīgi pieejama, un visas funkcionalitātes darbosies kā paredzēts.

Lietotājiem tiks nodrošināta iespēja reģistrēties, pierakstīties sistēmā, aplūkot agrāk veikto darbu kolekciju, pārskatīt savas pieprasītās uzdevumu sarakstu, modificēt profilu iestatījumus un iesniegt jaunus pieprasījumus.

Administratoriem tiks dotas iespējas izmantot saturpārvaldes sistēmu, pārskatīt un apstrādāt klientu dati un dizaina pieprasījumi.

Galīgais mērķis tiks sasniegts, kad tīmekļa vietnē nebūs nekādu būtisku kļūdu un tā būs pilnīgi pārbaudīta, lai nodrošinātu, ka tā ir gatava gan klientu, gan dizaina aģentūras lietošanai.

1. Programmatūras prasību specifikācija

Tālāk tiks izvērtētas projekta prasības. Tiks apskatīta produkta potenciālā darbības joma, sistēmas funkcionalitātes un citi aspekti, kas raksturo tās darbību. Turklāt tiks aplūkotas gan funkcionalitātes, gan nefunkcionālās prasības, kā arī gala lietotāju rakstur zīmes. Lai skaidrāk izprastu, tiks sniegti termini un saīsinājumu skaidrojumi.

* 1. Administrācija

Mērķis: Administrācija ir vieta, kur lietotāji var ielogoties, ievadot nepieciešamo informāciju attiecīgos ievades logos. Turklāt, ir iespējams arī veikt reģistrāciju, nospiežot "Reģistrēties" pogu. Šis funkcionalitātes risinājums ļauj lietotājiem gan pierakstīties sistēmā, gan izveidot jaunu kontu, nodrošinot pilnu piekļuvi mājaslapas funkcijām.

Administrācijas ievaddati:

|  |  |  |
| --- | --- | --- |
| Ievaddati | Obligāti | Piezīmes |
| Uzvārds | Jā |  |
| Parole | Jā |  |

Apstrāde: Nospiežot "Iesniegt" pogu, gan servera, gan lietotāja pusē, sistēma veic pārbaudi, vai visi lauki ir aizpildīti. Ja visi nepieciešamie lauki ir aizpildīti, servera puse veic papildu pārbaudes. Ja visas pārbaudes ir veiksmīgas un nav kļūdu, administrātors tiek savienots ar datu bāzi un veiksmīgi ielogojas sistēmā.

* 1. Lietotāja reģistrēšanās

Mērķis: Šī funkcija ļauj lietotājam reģistrēties ar lietotāja privilēģijām, kas atļauj piekļūt autorizēto lietotāju funkcijām un lapām. Pēc veiksmīgas reģistrācijas, lietotājs var pilnībā izmantot mājaslapas piedāvātās funkcijas, kas ir pieejamas tikai autorizētiem lietotājiem.

Ievaddati:

|  |  |  |
| --- | --- | --- |
| Ievaddati | Obligāti | Piezīmes |
| Vārds | Jā |  |
| Uzvārds | Jā |  |
| Parole | Jā |  |
| Epasts | Jā |  |
| Telefons | Jā |  |
| Dz.Datums | Jā |  |
| Adrese | Jā |  |

Apstrāde: Nospiežot "Reģistrēties" pogu, gan servera, gan lietotāja pusē, sistēma veic pārbaudi, vai visi lauki ir aizpildīti. Ja visi nepieciešamie lauki ir aizpildīti, servera puse veic pārbaudi, vai norādītais e-pasts jau nav reģistrēts sistēmā. Kad visi pārbaudes ir veiksmīgas un nav kļūdu, datu bāzē tiek ģenerēti visi nepieciešamie ievades dati no servera puses, kā arī lietotāja ievades dati. Pēc tam tiek veikta informācijas apkopošana un izveidots jauns lietotājs sistēmā.

* 1. Lietotāja pieslēgšanās

Mērķis: Nospiežot "Pieslēgties" pogu, gan servera, gan lietotāja pusē, sistēma veic pārbaudi, vai visi nepieciešamie lauki ir aizpildīti. Ja visi lauki ir aizpildīti, servera puse veic pārbaudi, vai ievadītās autentifikācijas informācijas ir derīga un atbilst esošiem datiem. Kad visas pārbaudes ir veiksmīgas un nav kļūdu, lietotājs tiek pieslēgts un iegūst pilnīgu piekļuvi lapas funkcijām, kas ir pieejamas tikai autorizētiem lietotājiem.

Ievaddati:

|  |  |  |
| --- | --- | --- |
| Ievaddati | Obligāti | Piezīmes |
| Vārds | Jā |  |
| Parole | Jā |  |

Apstrāde: Nospiežot "Iesniegt" pogu, gan servera, gan lietotāja pusē, sistēma veic pārbaudi, vai visi lauki ir aizpildīti. Ja visi nepieciešamie lauki ir aizpildīti, servera puse veic papildu pārbaudes, lai nodrošinātu datu kvalitāti un drošību. Ja visas pārbaudes ir veiksmīgas un nav kļūdu, administrātors tiek savienots ar datu bāzi un veiksmīgi ielogojas sistēmā, iegūstot pilnīgu piekļuvi un atļauju pārvaldīt lapas administratīvās funkcijas.

* 1. Problēma

Mērķis: Mērķis ir nodrošināt lietotājiem iespēju pievienot savas problēmas saistībā ar IT jomu. Šī funkcionalitāte ļauj lietotājiem reģistrēties sistēmā, veikt pievienošanu un sniegt detaļas par sastapto problēmu vai kļūdu. Mērķis ir nodrošināt efektīvu un vienkāršu veidu, kā lietotāji varētu sniegt informāciju par saviem IT jomas uzdevumiem un saņemt atbalstu no atbilstošajiem speciālistiem vai IT atbalsta komandu.

Ievaddati:

|  |  |  |
| --- | --- | --- |
| Ievaddati | Obligāti | Piezīmes |
| Apraksts | Jā |  |

Apstrāde: Nospiežot "Iesniegt" pogu, gan servera, gan lietotāja pusē, sistēma veic lauka pārbaudi. Ja lauks ir pareizi aizpildīts, servera pusē notiek papildu pārbaude. Kad visi pārbaudes ir veiksmīgi pabeigtas, bez jebkādām kļūdām, datu bāzē tiek ģenerēts un saglabāts nepieciešamais ievades dati no servera puses un lietotāja ievades. Pēc tam tiek veikta informācijas apkopošana un izveidots jauns IT atbalsta pieprasījums, ko administratori var apskatīt, kā arī attiecīgais lietotājs.

1. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums
   1. Izvēlēto risinājumu līdzekļu un valodu apraksts

Projekta izstrādē tika izmantoti daudzi un dažādi rīki un risinājumi, ņemot vērā projektā iesaistīto komplekso sistēmu. Tika izvēlēti un pielietoti līdzekļi, kas atbilst projekta prasībām un kompleksitātei, ievērojot esošo infrastruktūru. Dažādas valodas un tehnoloģijas tika izmantotas, lai nodrošinātu veiksmīgu projekta realizāciju un integrāciju ar esošajiem risinājumiem.

Izstrādes rīki:

* Front-End izstrādei tika izvēlēts Visual Studio Code (IDE) sakarā ar tā plašo funkcionalitāti un izplatību, jo tas ir viens no vispopulārākajiem un visizmantotākajiem IDE pasaulē. Papildus tam, Visual Studio Code piedāvā daudz iespēju, piemēram, dažādu paplašinājumu lejupielādi, kas atbilst specifiskām vajadzībām, kā arī ļauj mainīt izskatu pēc personīgām preferencēm. Šis rīks tika izvēlēts, ņemot vērā tā plašās funkcionalitātes priekšrocības un pielāgojamību lietotāja vajadzībām.
* Projekta izstrādes laikā tika izmantots Git un GitHub kā versiju kontrolēšanas sistēma, kas nodrošina aizsardzību pret datu zudumiem un ļauj veikt efektīvu versiju kontrolēšanu. GitHub piedāvā papildu priekšrocības, piemēram, iespēju apskatīt iepriekšējās izmaiņas un fiksēt problēmas vai jautājumus saistībā ar projektu, kas padara to ērtu un noderīgu rīku projekta vadībai.

Dizaina rīki:

* **Adobe Photoshop**, kas ir grafiskā dizaina rīks, ar kura palīdzbu var veikt ļoti dažādas grafiskā dizaina darbus. Šis rīks tika izmantots paša projekta sākumā, lai uzskicētu mājasapas prototipu ar kura palīdzību pēc tam varētu veidot noteiktās sadaļas un lapas.

Projekta izstrādē tika izmantotas valodas, kas jau iepriekš bija izmantotas uz izmantotās sistēmas, nodrošinot sadarbību ar esošiem risinājumiem un rīkiem, kā arī optimizējot izstrādes procesu. Izmantotās valodas ietvēra HTML, CSS, JavaScript, PHP un SQL, kur katrai valodai bija savs mērķis un funkcija. HTML tika izmantots, lai definētu mājaslapas struktūru, CSS - pielāgotu izskatu, JavaScript - nodrošinātu interaktivitāti, PHP - veiktu servera puses datu apstrādi, un SQL - veiktu datu bāzes darbības.

Front-End valodas:

Projekta izstrādē tika izmantota HTML valoda, kas ir pamata valoda Front-End izstrādei. HTML ļauj definēt mājaslapas struktūru un komponentus, izmantojot atzīmes un elementus. Tas nodrošina informācijas hierarhiju un saziņu ar citām valodām un resursiem. HTML ir viena no biežāk izmantotajām valodām Front-End attīstības jomā, tāpēc tā tika izvēlēta, lai nodrošinātu pareizu un saskaņotu mājaslapas attēlošanu un struktūru.

Projekta izstrādē tika izmantota CSS valoda (Cascading Style Sheets), kas ir bieži lietota valoda Front-End attīstībā. CSS nodrošina stilu un izkārtojumu definīciju mājaslapai, ļaujot pielāgot tās izskatu un dizainu. Ar CSS palīdzību var kontrolēt fontus, krāsas, izmērus, atkāpes un daudzas citas vizuālās īpašības. Tās mērķis ir atdalīt dizainu no satvara, nodrošinot pārskatāmu un viegli pielāgojamu stilu sistēmu. Izmantojot CSS, var radīt pievilcīgu un konsistentu izskatu visām mājaslapas lapām un elementiem.

Back-End valodas:

Projekta izstrādē tika izmantota PHP valoda (Hypertext Preprocessor), kas ir plaši izmantota servera puses programmēšanas valoda. PHP ir dinamiska un elastīga valoda, kas ļauj veidot interaktīvas un dinamiskas mājaslapas. Tā ir spējīga apstrādāt datus, sazināties ar datubāzēm, apstrādāt formas ievades un veikt citas servera puses darbības. PHP valoda ir atvērta un viegli integrējama citās tehnoloģijās, tādēļ to bieži izmanto mājaslapu un web lietotņu attīstībā. Ar PHP var veidot gan vienkāršas mājaslapas, gan kompleksas web aplikācijas, sniedzot plašas iespējas funkcionalitātes un datu apstrādes ziņā.

* 1. Iespējamo (*alternatīvo)* risinājuma līdzekļu un valodu apraksts

Iespējamās alternatīvas tiks pamatotas uz esošiem instrumentiem un programmēšanas valodām, kas jau ir pieejami. Šīs alternatīvas tiek apsvērtas, ņemot vērā to piemērotību visiem nepieciešamajiem uzdevumiem un projektam izvirzītajām prasībām.

Alternatīvas Front-End:

* HTML alternatīva React **–** React tiek uzskatīts par alternatīvu HTML izstrādei dažādos aspektos. React piedāvā plašākas un dinamiskākas iespējas interaktīvu un modulāru lietotāja saskarsmes izveidē, ļaujot izveidot atkārtojamus komponentus, efektīvi atjaunināt vietni un piesaistīt JavaScript bibliotēkas. Tādējādi React nodrošina programmētājiem rīkus un efektīvāku veidu, kā attīstīt modernas un reaktīvas vietnes. Tomēr ir jāatzīmē, ka HTML joprojām ir būtiska sastāvdaļa, kas nodrošina vietnes struktūru un saturu.
* CSS alternatīva **SASS** – SASS ir populāra CSS preprocesora valoda. CSS ir daudz versiju, kas var radīt neskaidrību. SASS palīdz rakstīt tīru un vieglu CSS kodu. SASS ļauj sarakstīt mazāku kodu nekā CSS, ja ir nepieciešams veikt vienādu uzdevumu.

Alternatīvas Back-End:

* Python ir augstas līmeņa programmēšanas valoda, kas ir pazīstama ar savu lasāmību un vienkāršību. Tā piedāvā plašas bibliotēkas un ietvarus, kas atvieglo programmēšanas procesu. Python ir universāla valoda, kas spēj apstrādāt dažādus uzdevumus, sākot no datu analīzes līdz tīmekļa izstrādei un mākslīgajai intelektam. Ar tās plašo atbalstu un aktīvo kopienas atbalstu, Python ir kļuvis par vienu no populārākajām programmēšanas valodām.

1. Sistēmas modelēšana un projektēšana
   1. Sistēmas struktūras modelis

Informācijas sistēmas struktūras modelis ir veids, kā sistemātiski organizēt un veidot informācijas sistēmu, iekļaujot dažādus sastāvdaļas, to attiecības un hierarhiju. Tas nodrošina holistisku skatījumu uz sistēmas komponentiem, to savstarpējo mijiedarbību un organizāciju jau pirms projekta izstrādes sākuma. Informācijas sistēmas struktūras modelis sniedz plašāku priekšstatu par komponentu savstarpējo mijiedarbību un sistēmas organizēšanu, tādējādi palīdzot izstrādāt detalizētu sistēmas izstrādes plānu un identificēt nepieciešamās sastāvdaļas.

Komponenti:

* Lietotāju saskarne: Nodrošina lietotājiem iespēju veikt reģistrāciju, pieslēgties, pievienot pieprasījumus par IT Atbalstu un apskatīt savus veidotos pieprasījumus. Galvenais fokuss šajā komponentā ir ērti un intuitīvi izmantojama saskarne, kas atvieglo lietotāju darbības.
* Administratora paneļi: Nodrošina administratoriem papildu iespējas un funkcijas, piemēram, piekļuvi satura pārvaldības sistēmai, rediģēšanas un pārvaldības rīkiem, kā arī dažādu lietotāju grupu pārvaldību. Administrātoriem tiek sniegta paplašināta kontrole un iespējas, lai veiktu nepieciešamos uzdevumus, vadītu satura saturu un efektīvi pārvaldītu lietotāju grupas.

Attiecības:

* Lietotāju saskarne un datubāze: Lietotāju saskarne sazinās ar datubāze, lai saglabātu un iegūtu informāciju par lietotājiem, pieprasījumiem un dizainiem. Šī savstarpējā attiecība nodrošina efektīvu datu uzturēšanu un atjaunošanu sistēmā, ļaujot glabāt un piekļūt nepieciešamajiem datiem.

Hierarhija:

* Mājaslapas komponenti: Sistēmas struktūra ir iedalīta hierarhiskā veidā, kur galvenās sastāvdaļas ir lietotāju saskarne un administrators panelis. Šie elementi pārstāv augstāko līmeni un ietver daudzus zemākas līmeņa komponentus un moduļus, kuri veic konkrētas funkcijas un uzdevumus sistēmā.
  1. ER diagramma

ER diagramma( sk. 4.2 att.).

Attēls, kurā ir teksts, ekrānuzņēmums, fonts, cipars

Apraksts ģenerēts automātiski

4.2. att.

* 1. Funkcionālais un dinamiskais sistēmas modelis

Šie modeļi, kas tiek izmantoti sistēmas projektēšanā, sniedz skaidru struktūru un izpratni par sistēmas funkcionalitāti un datu plūsmu. Tie palīdz vizualizēt lietotāju darbības un sistēmas darbības gaitu, nodrošinot izpratni par to, kā dati tiek apstrādāti un pārvietoti sistēmā. Šie modeļi, piemēram, lietotāju darbības diagrammas vai procesu plūsmas diagrammas, ir noderīgi rīki, lai nodrošinātu precīzu un saskaņotu izpratni par sistēmas darbību gan attīstītājiem, gan citiem iesaistītajiem.

Funkcionālais modelis:

* Lietotāju pieteikšanās un reģistrācija: Lietotājiem ir iespēja reģistrēties sistēmā, ievadot savu lietotājvārdu un paroli. Ja vēl nav konta, lietotāji var izveidot jaunu kontu, kuram tiks piešķirts unikāls identifikators. Pēc veiksmīgas reģistrācijas lietotāji var izmantot savus pieejas datus, proti, lietotājvārdu un paroli, lai pieslēgtos sistēmai un piekļūtu savam kontam.
* Problēmu apskate: Lietotājiem ir iespēja apskatīt savu jau izveidoto problēmu statusu, lai iegūtu informāciju par noteikto dizainu. Viņi var pārbaudīt, vai viņu problēma ir atrisināta vai vēl tiek darīts pie dizaina procesa. Šī iespēja nodrošina lietotājiem pārskatu par to, kādi pasākumi ir veikti un kāds ir pašreizējais progress, attiecībā uz viņu pieprasīto dizainu. Tas sniedz viņiem informāciju un skaidrību par to, kurā stadijā atrodas viņu problēmas risināšana.
* IT Atbalsta pieprasījumu pievienošana: Lietotāji var iesniegt IT atbalsta pieprasījumus, norādot savas nepieciešamības un prasības, lai saņemtu atbilstošu palīdzību attiecībā uz IT jomu. Šī iespēja ļauj lietotājiem precīzi aprakstīt savas problēmas un sniegt detalizētu informāciju, kas ļauj IT atbalsta komandai efektīvi risināt pieprasījumus.

Dinamiskais modelis:

* Datu pārvaldība un saglabāšana: Sistēma nodrošina informācijas saglabāšanu un pārvaldību par lietotājiem, problēmām, pieprasījumiem un citiem saistītiem datiem datubāzē. Tas ļauj efektīvi organizēt un uzturēt informāciju, kas nepieciešama sistēmas darbībai, lietotāju atbalstam un datu analīzei.
* Lietotāju atļaujas un piekļuves kontrole: Sistēma nodrošina dažādas piekļuves tiesības lietotājiem un administratoriem, ierobežojot noteiktas funkcijas un lapas konkrētām lietotāju grupām. Tas ļauj efektīvi pārvaldīt sistēmas piekļuvi un nodrošināt, ka katram lietotājam ir atbilstošas funkcionalitātes un informācijas pieejamība, atkarībā no viņu lomām un atbildības.
  1. Aktivitāšu diagramma

Aktivitāšu diagramma(Skat. 4.4.un 4.5 att.)

Attēls, kurā ir diagramma, plāns, rasējums, teksts

Apraksts ģenerēts automātiski

4.5. att.

Attēls, kurā ir diagramma, rasējums, plāns, rinda

Apraksts ģenerēts automātiski

4.5. att.

* 1. Lietotājgadijūma diagramma ( Use Case )

Aktivitāšu diagramma(Skat. 4.5. att.)

Attēls, kurā ir diagramma, teksts, rasējums, plāns

Apraksts ģenerēts automātiski

4.5. att.

1. Sistēmas moduļu apraksts un algoritmu shēmas

Lietotāju pieteikšanās un reģistrācijas modulis:

* Šis modulis piedāvā lietotāju reģistrācijas un pieslēgšanās funkcionalitāti sistēmā. Tas ietver formu un darbības, kas saistītas ar lietotāju autentifikāciju un autorizāciju, ļaujot lietotājiem izveidot jaunus kontus un pieslēgties ar esošiem kontiem. Šāds mehānisms nodrošina drošu un kontrolētu piekļuvi sistēmai, aizsargājot lietotāju datus un ierobežojot sistēmā veicamo darbību apjomu atkarībā no lietotāja atļaujām.

IT Atbalsta pieprasīšanas modulis:

* Šis modulis ir atbildīgs par IT Atbalsta pieprasījumu pievienošanu un rediģēšanu. Tas nodrošina funkcionalitāti, kas ļauj lietotājiem pievienot jaunus pieprasījumus. Lietotājiem ir jāaizpilda specifiska forma, kurā jāievada visi nepieciešamie lauki ar attiecīgo informāciju. Šis process nodrošina sistēmas pareizu darbību, cenšoties nodrošināt, ka visi būtiskie dati ir ievadīti, lai varētu veiksmīgi izveidot un apstrādāt IT Atbalsta pieprasījumus.

Administrācija:

* Šis modulis nodrošina visu pieprasījumu apskatīšanu no datubāzes. Piekļuve šim modulim ir atļauta un paredzēta tikai administratora grupas lietotājiem. Tas tiek izmantots administratora panelī, lai rādītu visus pieprasījumus, kas ir iesniegti sistēmai. Administrators var apskatīt un pārvaldīt šos pieprasījumus, veicot attiecīgas darbības vai sniedzot atbildes uz tiem. Šis modulis nodrošina pārskatu par visiem pieprasījumiem vienkāršā un ērtā veidā, ļaujot administratoram efektīvi vadīt un risināt iesniegtos pieprasījumus.

1. Lietotāju ceļvedis

Lietotāja ceļvedis ir izstrādāts, lai sniegtu lietotājiem pamatinformāciju par sistēmas elementiem un galvenajām funkcijām, kas palīdzēs vieglāk orientēties sistēmā un saprast gan lietotāja, gan administratora specifiskās funkcijas. Lietotāja ceļvedī attēlotie dati ir izdomāti un nav reāli. Tos izmanto tikai informatīvi, lai demonstrētu funkciju darbību un funkciju nozīmi pareizai lietošanai. Attēlojamā informācija, attēli un teksts var tikt modificēti atbilstoši satura pārvaldības sistēmas datiem un klienta pieprasītajām dizaina izmaiņām.

* 1. Pieslēgšanās

Lai lietotājs varētu piekļūt specifiskām funkcijām, piemēram, dizaina pieprasījuma pievienošanai, ir nepieciešams, lai viņš pieslēgtos savam profilam. Pieslēdzoties profilam, lietotājam ir jāievada savs lietotāja vārds un parole, lai autentificētos sistēmā. Šī informācija nodrošina drošu un personīgu piekļuvi lietotāja specifiskajām funkcijām.

Lietotājam ir jāievada lietotāja vārds un parole divos atsevišķos laukos. Kad nepieciešamie lauki ir aizpildīti, lietotājam ir jānospiež "Pieslēgties" poga (skatīt 6.1.1.att.), kas veiks pārbaudi, vai ievadītie dati atbilst datubāzes informācijai. Šī pārbaude nodrošina lietotāja autentifikāciju un atļauj piekļūt sistēmas funkcijām.

* Ja ievadītie dati nesakrīt ar datubāzes datiem, piemēram, ja norādītais lietotāja vārds neeksistē vai ja ievadītajam lietotāja vārdam ir nepareiza parole, sistēma parādīs paziņojumu par nepareizu lietotāja vārdu vai paroli (skatīt 6.1.2.attēlu). Šis paziņojums informēs lietotāju, ka autentifikācija neizdevās un lūgs pārbaudīt ievadītos datus un mēģināt vēlreiz. Tas nodrošina drošību un aizsargā sistēmu no neautorizētas piekļuves.
* Ja ievadītie dati sakrīt ar datubāzes datiem, lietotājs tiks novirzīts uz lietotāja mājaslapu. Šis paziņojums apliecina, ka autentifikācija ir izdevusies un lietotājs ir veiksmīgi pieslēdzies sistēmai. Tādējādi lietotājs varēs pilnvērtīgi izmantot mājaslapas funkcijas un resursus, kas viņam ir pieejami pēc autorizācijas.

Attēls, kurā ir teksts, ekrānuzņēmums, aplis, Elektriski zila

Apraksts ģenerēts automātiski

6.1.1. att.

Attēls, kurā ir teksts, ekrānuzņēmums, bumba

Apraksts ģenerēts automātiski

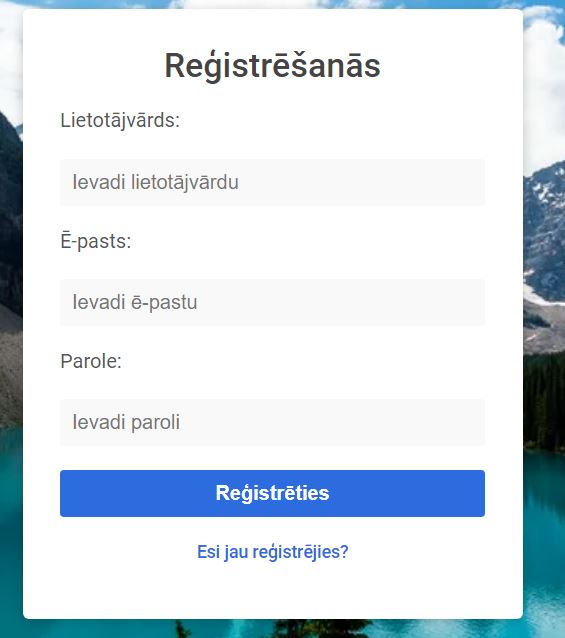
6.1.2. att.

* 1. Reģistrēšanās

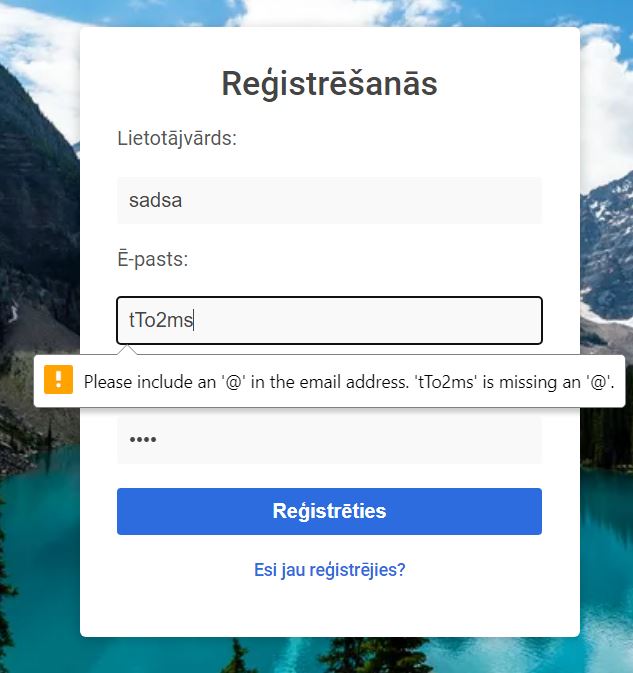
Lai lietotājs varētu piekļūt savam profilam un izmantot sistēmas funkcijas, viņam ir nepieciešams izveidot profilu. Sistēmā ir iekļauta profila izveides un reģistrācijas funkcionalitāte, kas ļauj lietotājam veikt savas personas informācijas ievadi un izveidot savu profilu. Profila izveide nodrošina lietotājam individuālu identitāti un piekļuvi personalizētām iespējām un resursiem sistēmā.

Lietotājam ir jāievada reģistrācijas dati septiņos laukos. Pirmajā laukā tiek ievadīts vārds, otrajā - uzvārds, trešajā - e-pasta adrese, ceturtajā - parole, piektajā - atkārtota parole, sestajā - telefona numurs, un septītajā - faktiskā adrese. Kad lietotājs nospiež pogu "Reģistrēties" (sk. 6.2.1.att.), šie dati tiek nosūtīti uz datubāzi, kur tie tiek saglabāti.

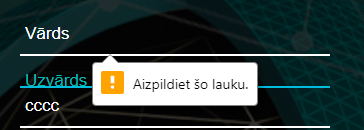
* Ja nav ievadīts lietotāja vārds un nospiesta poga ‘Reģistrēties’, tad tiks parādīts, ka ir nepieciešams aizpildīt šo lauku. ( sk. 6.2.3.att. ).
* Ja visi lauki ir ievadīti pareizi, bet ievadītais ē-pasts jau eksistē, tad tiks parādīts, ka ievadītais ē-pasts jau ir reģistrēts sistēma un reģistrācija neizdosies, jo ē-pasts ir unikāls. ( sk. 6.2.4.att. ).
* Ja visi lauki ir ievadīti pareizi un viss sakrīt ar servera prasībām, tad tiks attēlots, ka Reģistrācija ir izdevusies veiksmīgi. ( sk. 6.2.5.at ).



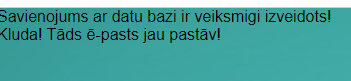
6.2.1. att.



6.2.2. att.



6.2.3. att.



6.2.4. att.

Attēls, kurā ir teksts, ekrānuzņēmums, programmatūra, operētājsistēma

Apraksts ģenerēts automātiski

6.2.5. att.

* 1. Navigācija

Mājaslapai ir trīs veidu navigācijas joslās, kas atvieglo lietotāju pārvietošanos un uzlabo lietojamību. Pieslēgušajiem lietotājiem, nepieslēgušajiem lietotājiem un administratoriem ir atsevišķas navigācijas joslas, kas piedāvā pielāgotu satura un funkciju pieejamību atbilstoši katras grupas vajadzībām.

* Lietotājiem, kuri nav pieslēgušies, navigācijas panelī ir pieejamas divas lapas - "Sākums" un "Kontakti". Lietotājiem ir arī iespēja izvēlēties vai pieslēgties savam kontam vai veikt reģistrāciju. ( sk. 6.3.1. att. ).
* Lietotājiem kuri ir pieslēgušies, navigācijas panelī ir iespēja aiziet uz lapām – sākums un kontakti. Lietotājam ir opcijas atslēgties, gan aiziet uz savu profilu. ( sk. 6.3.2. att. ).
* Administratoram navigācijas panelī ir tikai viens pieejams elements, kas ļauj apskatīt reģistrēto klientu skaitu, darbinieku skaitu un gan pabeigto, gan nepabeigto problēmu skaitu. Šis informācijas panelis sniedz apkopoto statistiku par sistēmas lietojumu un problēmu stāvokli. ( sk. 6.3.3. att. ).



6.3.1. att.



6.3.2. att.



6.3.3. att.

* 1. Sākums

Sākuma lapā lietotāji var iepazīties ar pamatinformāciju par aģentūru un izmantot opciju "Pievienoties" sākuma sekcijā. Turklāt, sākuma lapā ir pieejama arī iespēja veikt reģistrāciju. Ja lietotājs izvēlas reģistrēties, viņš var nospiežot pogu "Reģistrēties", uzsākt reģistrācijas procesu.

* Nospiežot pogu ‘Pievienoties, lietotājs tiks aizvest uz ielogošanās lapu. ( sk. 6.4.1. att.).
* Nospiežot pogu ‘Reģistrēties, lietotājs tiks aizvest uz reģistrācijas lapu. ( sk. 6.4.2. att.).

Attēls, kurā ir teksts, fonts, ekrānuzņēmums, Elektriski zila

Apraksts ģenerēts automātiski

6.4.1. att.

Attēls, kurā ir teksts, ekrānuzņēmums, fonts, logotips

Apraksts ģenerēts automātiski

6.4.2. att.

* 1. Problēmas pievienošana

Lietotājam, kas ir pieslēdzies, ir iespēja pieprasīt IT atbalstu, ko administratori varēs apskatīt. Lai pieprasītu IT atbalstu, lietotājam ir jāaizpilda lauks, kurā jāapraksta problēma. Visi lauki ir jāaizpilda, lai veiksmīgi pievienotu problēmu.

Lietotājam ir jāaizpilda viens ievades lauks - apraksts. Kad visi ievades lauki ir aizpildīti, lietotājs var nospiegt pogu "Sūtīt" (sk. 6.8.1. attēlu). Pēc pogas nospiešanas tiks pārbaudīts, vai visi lauki ir aizpildīti un vai ievadītā informācija atbilst servera kritērijiem.

* Lietotājam ir obligāti jāaizpilda visi ievades lauki ar datiem, kas nepieciešami IT atbalstam (sk. 6.8.2. attēlu).
* Ja lietotājs nospiež pogu "Iesniegt", bet ievades lauks "Saturs" nav aizpildīts, tiks parādīts paziņojums, ka satura lauks ir obligāti jāaizpilda (sk. 6.8.3. attēlu).
* Ja lietotājs nospiež pogu "Iesniegt" un visi ievadītie lauki ir pareizi aizpildīti, dati tiks veiksmīgi nosūtīti uz serveri, un tiks parādīts paziņojums, ka dizaina pieprasījums ir veiksmīgi izveidots (sk. 6.8.4. attēlu).



6.8.1. att.

Attēls, kurā ir teksts, ekrānuzņēmums, zīmols, ūdens

Apraksts ģenerēts automātiski

6.8.2. att.

Attēls, kurā ir teksts, fonts, ekrānuzņēmums, logotips

Apraksts ģenerēts automātiski

6.8.3. att.

Attēls, kurā ir teksts, ekrānuzņēmums, zili zaļš, fonts

Apraksts ģenerēts automātiski

6.8.4. att.

* 1. Lietotāja profils

Lietotāju profils ir būtiska sastāvdaļa mājaslapā, nodrošinot katram lietotājam individuālu unikālu pieredzi. Profilā lietotājs var apskatīt savas pievienotās problēmas vai jautājumus, kas ir saistīti ar mājaslapu. Šis profila elements ir paredzēts, lai sniegtu lietotājam pilnīgu pārskatu par viņa darbībām un interakciju ar mājaslapu, tādējādi veicinot personalizētu un efektīvu lietotāja pieredzi. ( sk. 6.9.1. att. ).

Lietotājiem ir iespēja apskatīt pilnu sarakstu ar saviem IT problēmām, kur katrai problēmai ir norādīts nosaukums, kategorija un statuss. Šis saraksts sniedz lietotājiem pārskatu par visām viņu iesniegtajām problēmām un ļauj viegli identificēt un izsekot katrai no tām. Papildus tam, lietotāji var iegūt papildinformāciju par katru problēmu, kas palīdzēs labāk saprast un pārvaldīt tos. ( sk. 6.9.1. att. ).

* Lietotājam ir iespēja savā profila lapā izmantot pogu "Pievienot", kas noved lietotāju uz problēmu pieprasījuma pievienošanas lapu. Šī poga nodrošina ērtu veidu, kā lietotāji varētu pievienot jaunas problēmas vai pieprasījumus, sniedzot nepieciešamo informāciju. Tas vienkāršo procesu un ļauj lietotājiem viegli reģistrēt un ziņot par problēmām, ar kurām tie saskaras. ( sk. 6.9.2. att. ).
* Lietotāja specifiskais problēma ir pabeigta un administrācija ir nomainījusi statusu uz ‘Izpildīts’ ( sk. 6.9.3. att. ).
* Lietotājs ir pievienojis problēmas pieprasījumu un noteiktas darbs nav ticis aiztikts no administrācijas puses, kas nozīme, ka darba status ir ‘Nav pabeigts’. ( sk. 6.9.4. att. ).

Attēls, kurā ir teksts, ekrānuzņēmums, fonts, programmatūra

Apraksts ģenerēts automātiski

6.9.1. att.

Attēls, kurā ir teksts, fonts, ekrānuzņēmums, logotips

Apraksts ģenerēts automātiski

6.9.2. att.

Attēls, kurā ir teksts, fonts, ekrānuzņēmums

Apraksts ģenerēts automātiski

6.9.3. att.

Attēls, kurā ir teksts, ekrānuzņēmums, fonts

Apraksts ģenerēts automātiski

6.9.4. att.

* 1. Administrācijas panelis

Administrācijas panelis ir izveidots mājaslapas administratoriem, lai tie varētu pārvaldīt IT atbalsta pieprasījumus. Šajā panelī administratori var apskatīt visus lietotāju iesniegtos pieprasījumus un veikt nepieciešamos pasākumus. Viņi ir pilnvaroti mainīt problēmu statusus, kā arī iegūt detalizētu informāciju par katru problēmu. ( sk. 6.10.1. att. ).

Administrācijas panelim var piekļūt tikai lietotāji ar ‘Admin’ grupu. Ja lietotājs, kuram nav admin grupa meiģina atvērt administrācijas paneli, tad interneta pārlūks parādīts lietotājam ka ir nepareizs lietotāja vārds vai parole.

Administratora panelī ir pieejams pilns saraksts ar visiem dizaina pieprasījumiem, kur administratori var apskatīt informāciju par katru pieprasījumu, tostarp vārdu, problēmu un statusa maiņu. Attēlā 6.10.1. ir parādīts šis administrācijas panelis, kurā ir redzama visaptveroša informācija par katru dizaina pieprasījumu un iespēja veikt nepieciešamos pasākumus attiecībā uz katru no tiem.

* Administratoriem ir pilnīga iespēja pārskatīt visus lietotāju pievienotās problēmas un izvēlēties atbilstošās funkcijas un pogas. Viņi var rūpīgi izvērtēt katru problēmu, apskatīt to detalizēto informāciju un veikt nepieciešamās darbības, lai risinātu vai noslēgtu konkrētas problēmas. ( sk. 6.10.1. att. ).
* Ja administratori ir atvēruši administrāciju, tad viņiem ir opcija nospiest pogu ‘izpildīts’ (sk. 6.10.2. att. ), kas nomainīs problēmas statusu uz ‘Pabeigts’.

Attēls, kurā ir teksts, ekrānuzņēmums, fonts

Apraksts ģenerēts automātiski

6.10.1. att.

Attēls, kurā ir teksts, ekrānuzņēmums, fonts, grafika

Apraksts ģenerēts automātiski

6.10.2. att.

1. Testēšanas dokumentācija

Tika veikta mājaslapas testēšana, lai pārbaudītu tās darbību un atklātu potenciālās kļūdas un problēmas pirms tās publiskās pieejamības. Testēšanas galvenais mērķis bija nodrošināt, ka mājaslapa darbojas pareizi un sasniedz paredzēto rezultātu visiem lietotājiem. Testēšanai tika izmantota melnās kastes testēšanas metode, kurā tika pārbaudīta mājaslapas funkcionālitāte neatkarīgi no iekšējās realizācijas detaļām.

* Melnās kastes testēšanas metode ir veids, kā testēt programmu, nepazīstot tās iekšējo kodu vai realizāciju. Šo metodi var izmantot arī cilvēki, kuriem nav programēšanas vai sistēmu uzbūves zināšanas. Melnās kastes testēšanas mērķis ir pārbaudīt programmu, izmantojot publiski pieejamo specifikācijas informāciju un datu ievades. Galvenais mērķis ir pārbaudīt, vai mājaslapas rezultāts un darbība atbilst sagaidītajam.
  1. Melnās kastes testēšanas piemēri

Mājaslapas testēšans piemēri tiks sadalīti un pa mājaslapas lapām, lai testēšana būtu vieglāk pārskatāma un lasāma.

* 1. Navīgacijas jolsa ( Nepieslēdzies lietotājs )

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testēšanas identifikātora numurs | Lapas nosaukums | Darbība | Solis | Gaidītais rezūltāts | Faktiskais rezūltāts | Kļūdas apraksts |
| #1 | index | Tiek nospiests kreisas peles klikšķis uz ‘Sākums’ pogu | 1 | Mājaslapa tiek atsvaidzināta un lietotājs paliek sākuma lapā | Mājaslapa tiek atsvaidzināta un lietotājs paliek sākuma lapā |  |
| #2 | index | Tiek nospiests kreisas peles klikšķis uz ‘Kontakti’ pogu | 1 | Lietotājs teik aizvest uz Kontakti lapu | Poga nereaģē. |  |
| #3 | index | Tiek nospiests kreisas peles klikšķis uz ‘Pievienoties’ pogu | 1 | Lietotājs tiek aizvest uz pieslēgšanās lapu | Lietotājs tiek aizvest uz pieslēgšanās lapu |  |
| #4 | index | Tiek nospiests kreisas peles klikšķis uz ‘Reģistrēties’ pogu | 1 | Lietotājs tiek aizvest uz reģistrēšanās lapu | Lietotājs tiek aizvest uz reģistrēšanās lapu |  |
| #5 | index | Tiek nospiests kreisas peles klikšķis uz ‘Pieslēgties’ pogu | 1 | Lietotājs tiek aizvest uz pieslēgšanās lapu | Lietotājs tiek aizvest uz pieslēgšanās lapu |  |

* 1. Navigācijas josla ( Pieslēdzies lietotājs )

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testēšanas identifikātora numurs | Lapas nosaukums | Darbība | Solis | Gaidītais rezūltāts | Faktiskais rezūltāts | Kļūdas apraksts |
| #1 | index | Tiek nospiests kreisas peles klikšķis uz ‘Sākums’ pogu | 1 | Mājaslapa tiek atsvaidzināta un lietotājs paliek sākuma lapā | Mājaslapa tiek atsvaidzināta un lietotājs paliek sākuma lapā |  |
| #2 | index | Tiek nospiests kreisas peles klikšķis uz ‘Kontakti’ pogu | 1 | Lietotājs tiek aizvest uz Kontakti lapu | Poga nereaģē. |  |
| #3 | index | Tiek nospiests kreisas peles klikšķis uz ‘logout’ pogu | 1 | Lietotājs tiek izlogots no sesijas. | Lietotājs tiek izlogots no sesijas |  |
| #4 | index | Tiek nospiests kreisas peles klikšķis uz ‘Mans profils’ pogu | 1 | Lietotājs tiek aizvest uz sava profila lapu | Lietotājs tiek aizvest uz sava profila lapu |  |
| #5 | index | Tiek nospiests kreisas peles klikšķis uz ‘Pievienoties’ pogu | 1 | Lietotājs tiek aizvest uz pieslēgšanās lapu | Lietotājs tiek aizvest uz pieslēgšanās lapu |  |

7.3 Lietotāja lapa

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testēšanas identifikātora numurs | Lapas nosaukums | Darbība | Solis | Gaidītais rezūltāts | Faktiskais rezūltāts | Kļūdas apraksts |
| #1 | lietotajs | Tiek nospiests kreisas peles klikšķis uz ‘Sākums’ pogu | 1 | Lietotājs tiek aizvests uz sākuma lapu | Lietotājs tiek aizvests uz sākuma lapu |  |
| #2 | lietotajs | Tiek nospiests kreisas peles klikšķis uz ‘Kontakti’ pogu | 1 | Lietotājs tiek aizvest uz Kontakti lapu | Poga nereaģē. |  |
| #3 | lietotajs | Tiek nospiests kreisas peles klikšķis uz ‘logout’ pogu | 1 | Lietotājs tiek izlogots no sesijas. | Lietotājs tiek izlogots no sesijas |  |
| #4 | lietotajs | Tiek nospiests kreisas peles klikšķis uz ‘Mans profils’ pogu | 1 | Mājaslapa tiek atsvaidzināta un lietotājs paliek lietotāja lapā | Lietotājs tiek aizvest uz sava profila lapu |  |
| #5 | lietotajs | Tiek nospiests kreisas peles klikšķis uz ‘Pievienot’ pogu | 1 | Lietotājs tiek aizvest uz addproblem lapu | Lietotājs tiek aizvest uz addproblem lapu |  |

**7.4** Lietotāja reģistrēšanās lapa

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testēšanas identifikātora numurs | Lapas nosaukums | Darbība | Solis | Gaidītais rezūltāts | Faktiskais rezūltāts | Kļūdas apraksts |
| #1 | userSignUP | Tiek nospiests kreisas peles klikšķis uz ‘Atgriezties uz sākumu’ pogu | 1 | Lietotājs tiek aizvests uz sākuma lapu | Lietotājs tiek aizvests uz sākuma lapu |  |
| #2 | userSignUP | Tiek aizpildīta visa nepieciešamā reģistrēšanās informācija un nospiests kreisas peles klikšķis uz ‘Reģistreties’ pogu | 1 | Lietotājs tiek aizvest uz pieslēgšanās lapu | Lietotājs tiek aizvest uz pieslēgšanās lapu |  |

**7.5** Pieslēgšanās lapa

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testēšanas identifikātora numurs | Lapas nosaukums | Darbība | Solis | Gaidītais rezūltāts | Faktiskais rezūltāts | Kļūdas apraksts |
| #1 | userLogin | Tiek nospiests kreisas peles klikšķis uz ‘Sākums’ pogu | 1 | Lietotājs tiek aizvests uz sākuma lapu | Lietotājs tiek aizvests uz sākuma lapu |  |
| #2 | userLogin | Tiek nospiests kreisas peles klikšķis uz ‘Reģistrēties’ pogu | 1 | Lietotājs tiek aizvest uz reģistrēšanās lapu | Lietotājs tiek aizvest uz reģistrēšanās lapu |  |
| #3 | userLogin | Tiek nospiests kreisas peles klikšķis uz ‘Pieslēgties’ pogu | 1 | Lietotājs tiek aizvest uz lietotājs lapu | Lietotājs tiek izlogots no sesijas |  |

**7.6** Problēmas iesniegšanas lapa

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testēšanas identifikātora numurs | Lapas nosaukums | Darbība | Solis | Gaidītais rezūltāts | Faktiskais rezūltāts | Kļūdas apraksts |
| #1 | addProblem | Tiek nospiests kreisas peles klikšķis uz ‘Send’ pogu | 1 | Lietotājs tiek aizvest uz lietotājs lapu un tiek parādīts iesniegtā problēma. | Lietotājs tiek aizvest uz lietotājs lapu un tiek parādīts iesniegtā problēma. |  |

1. Licences
   1. GNU Ģenerālā Publiskā Licence (GNU GPL)

* MIT Licence ir viena no vienkāršākajām atvērtā koda licencēm, kas nodrošina lietotājiem plašas brīvības izmantot programmatūru.
* Ar MIT licenci ir atļauts izmantot, modificēt un izplatīt programmatūru, gan komerciālos, gan nelabumos nolūkos, bez prasības publiskot izmaiņu avotkodu.
* MIT Licence nodrošina lietotājiem lielāku elastību un mazāku ierobežojumu attiecībā uz izplatīšanu un avotkoda atklātību, salīdzinot ar GNU GPL.
  1. MIT Licence
* MIT Licence ir viena no vienkāršākajām atvērtā koda licencēm, kas nodrošina lietotājiem plašas brīvības izmantot programmatūru.
* Ar MIT licenci ir atļauts izmantot, modificēt un izplatīt programmatūru, gan komerciālos, gan nelabumos nolūkos, bez prasības publiskot izmaiņu avotkodu.
* MIT Licence nodrošina lietotājiem lielāku elastību un mazāku ierobežojumu attiecībā uz izplatīšanu un avotkoda atklātību, salīdzinot ar GNU GPL.
  1. Salīdzinājums

Galvenais atšķirības punkts starp šīm divām licencēm ir prasība publiskot izmaiņu avotkodu. Ja vēlaties nodrošināt programmatūras brīvu izmantošanu, modificēšanu un izplatīšanu bez ierobežojumiem, bet neatkarīgi no tā, vai izmaiņu avotkods ir publiski pieejams, MIT Licence būtu laba izvēle. Ja vēlaties iekļaut programmatūru atvērtās kopienas attīstībā un garantēt, ka izmaiņu avotkods ir pieejams visiem, GNU GPL varētu būt piemērota izvēle.

* 1. Piemērotākā licence

MIT Licence ir piemērotāka izvēle manai IT atbalsta mājas lapai, jo tā nodrošina lietotājiem plašas brīvības izmantot un pielāgot programmatūru, neprasa publiskot izmaiņu avotkodu, un ļauj saglabāt programmatūru privātu. Tādējādi jūs varat veikt pielāgojumus pēc vajadzības, saglabājot pilnīgu kontroli pār programmatūras kodu un izmantojot to gan komerciāliem, gan nelabumos nolūkiem. Šī licences elastība un vienkāršība ir veicinājusi to populārākās izvēles statusu gan starp uzņēmējdarbības, gan atvērtās koda kopienas projektiem.

1. Programmatūras kods
   1. Administratora ielogošanās

<html lang="en-US">

<head>

<meta charset="utf-8">

<title>AdminLogin</title>

<link rel="stylesheet" href="https://fonts.googleapis.com/css?family=Open+Sans:400,700">

<link rel="stylesheet" href="login.css">

</head>

<?php

if (isset($\_POST['admin\_login'])) {

require("../connect.php");

session\_start();

$lietotaja\_vards = mysqli\_real\_escape\_string($savienojums, $\_POST['vards']);

$parole = mysqli\_real\_escape\_string($savienojums, $\_POST['parole']);

$sqlVaicajums = "SELECT \* FROM administracija WHERE lietotajvards = '$lietotaja\_vards'";

$rezultats = mysqli\_query($savienojums, $sqlVaicajums);

if (mysqli\_num\_rows($rezultats) == 1) {

while ($row = mysqli\_fetch\_array($rezultats)) {

if (password\_verify($parole, $row["parole"]) || ($parole == $row["parole"])) {

$\_SESSION["admin\_username"] = $lietotaja\_vards;

$\_SESSION["admin\_id"] = $row["darbinieki\_id"];

header("location:adminPanel.php");

} else {

echo "Nepareizs lietotāja vārds vai parole!";

}

}

} else {

echo "Nepareizs lietotaja vards vai parole!";

}

}

if (isset($GET['logout'])) {

session\_destroy();

}

?>

<div id="login">

<form name='form-login' method="post">

<input type="text" id="user" placeholder="Vārds" name="vards">

<input type="password" id="pass" placeholder="Parole" name="parole">

<input type="submit" value="Pieslēgties" name="admin\_login">

</form>

Dotais kods nodrošina administratora pieslēgšanos IT atbalsta mājaslapai, kur administrators var ievadīt savu lietotājvārdu un paroli, kas tiek pārbaudīti pret datubāzi, un ja tie ir derīgi, administrators tiek pāradresēts uz administrācijas paneli, bet ja tie nav derīgi, tiek izvadīts kļūdas ziņojums.

* 1. Administrātora panelis

<?php

$page = "panelis";

?>

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<link rel="stylesheet" href="admin\_styles.css">

<title>Document</title>

</head>

<body>

<?php

session\_start();

require("../connect.php");

if ($\_SESSION['admin\_username']) {

?>

<header>

<h3>Sveicināts <?php echo $\_SESSION['admin\_username']; ?>!</h3>

<hr>

</header>

<section>

<?php

$klientu\_skaitsSQL = "SELECT \* FROM klientu\_skaits";

$atrisinatas\_problemasSQL = "SELECT \* FROM atrisinatas\_problemas";

$problemu\_skaitSQL = "SELECT \* FROM problemu\_skaits";

$darbinieku\_skaitsSQL = "SELECT \* FROM darbinieku\_skaits";

$klientu\_skaits = mysqli\_query($savienojums, $klientu\_skaitsSQL);

$atrisinatas\_problemas = mysqli\_query($savienojums, $atrisinatas\_problemasSQL);

$problemu\_skaits = mysqli\_query($savienojums, $problemu\_skaitSQL);

$darbinieku\_skaits = mysqli\_query($savienojums, $darbinieku\_skaitsSQL);

?>

</div>

<div class="box">

<h4>Reģistrējušies klienti</h4>

<h4>Atrisinātās problemas</h4>

<h4>Problēmas kopā</h4>

<h4>Darbinieki</h4>

</div>

<div class="box">

<?php

while ($row = mysqli\_fetch\_array($klientu\_skaits)) {

echo "<h3 class = 'h3class'>{$row['skaits']}</h4>";

?>

<?php

}

while ($row = mysqli\_fetch\_array($atrisinatas\_problemas)) {

echo "<h3 class = 'h3class'>{$row['skaits']}</h4>";

?>

<?php

}

while ($row = mysqli\_fetch\_array($problemu\_skaits)) {

echo "<h3 class = 'h3class'>{$row['skaits']}</h4>";

?>

<?php

}

while ($row = mysqli\_fetch\_array($darbinieku\_skaits)) {

echo "<h3 class = 'h3class'>{$row['skaits']}</h4>";

?>

<?php

}

?>

</div>

</section>

<hr>

<h1>Neizdarītās problēmas</h1>

<section>

<?php

$problemaSQL = "SELECT lietotajs.vards,

lietotajs.lietotajs\_id, problema.apraksts, problema.problema\_id, problema.status

FROM problema

INNER JOIN lietotajs ON lietotajs.lietotajs\_id = problema.id\_lietotajs

WHERE problema.status = 'Nav pabeigts'";

$atlasaProblemas = mysqli\_query(

$savienojums,

$problemaSQL

) or die("Nekorekts vaicajums!");

if (mysqli\_num\_rows($atlasaProblemas) > 0) {

while ($row = mysqli\_fetch\_assoc($atlasaProblemas)) {

echo "

<div class='parent'>

<div class='box1'>

<h4>{$row['vards']} Problēma:</h4>

</div>

<div class='box1'>

<h4>{$row['apraksts']}

</h4>

</div>

<div class='box1'>

<form action = 'finishProblem.php' method = 'post'>

<button tpye = 'submit' name = 'izpildit'>Izpildīts</button>

</form>

</div>

</div>

<hr>

";

$\_SESSION["problemaID"] = $row["problema\_id"];

}

} else {

echo "Visas problēmas datubāzē ir izpildītas";

}

?>

</section>

<?php

} else {

header('Refresh: 0; url=adminLogin.php');

}

?>

</body>

</html>

Šis PHP koda gabals ir admina panelis, kas ielādē informāciju par reģistrētiem klientiem, problēmām un darbiniekiem, un parāda neatrisinātās problēmas, ļaujot adminam atzīmēt tās kā izpildītas.

* 1. Problēmas pievienošana.

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<title>Form Contact Us</title>

<link rel="stylesheet" href="userProblem.css">

</head>

<body>

<?php

session\_start();

if ($\_SESSION['username']) {

?>

<?php

$lietotajaID = $\_SESSION["id"];

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

require("connect.php");

if (isset($\_POST['iesniegt'])) {

$apraksts\_ievade = $\_POST['apraksts'];

if (!empty($apraksts\_ievade)) {

$registret\_problemu\_SQL = "INSERT INTO problema(apraksts, status, id\_lietotajs, id\_darbinieki)

VALUES ('$apraksts\_ievade', 'Nav pabeigts', '$lietotajaID', '1')";

if (mysqli\_query($savienojums, $registret\_problemu\_SQL)) {

echo "<div class = 'pazinojums zals'>Apsveicu! Problēma iesniegta!</div>";

header('Refresh: 1; url=lietotajs.php');

} else {

echo "<div class = 'pazinojums sarkans'>Radās kļūda! Problēmu ievietot neizdevās, kļūda sistēmā!</div>";

}

} else {

echo "<div class = 'pazinojums sarkans'>Problēmas ievietošana nav izdevusies! Ievades lauku problēmas!</div>";

}

}

} else {

}

?>

<div class="container">

<form method="post">

<h1>Problēmas pievienošana</h1>

<h4>Ievadiet problēmas aprakstu</h4>

<textarea required name="apraksts"></textarea>

<input type="submit" value="Send" id="button" name="iesniegt">

</form>

</div>

<?php

} else {

header('Refresh: 0; url=userLogin.php');

}

?>

</body>

</html>

Šis PHP un HTML kods nodrošina lietotājiem kontaktēšanās veidlapu, kur viņi var iesniegt problēmas aprakstu. Lietotājiem, kuri ir ielogojies, tiek parādīta forma, kurā viņi var aizpildīt problēmas apraksta lauku un iesniegt to. Kad forma ir iesniegta, dati tiek pārbaudīti un, ja apraksts ir norādīts, problēma tiek ievietota datubāzē. Pēc iesniegšanas, tiek parādīts atbilstošs paziņojums un lietotājs tiek novirzīts uz sākuma lapu.

* 1. Ielogošanās profilā

<!DOCTYPE html>

<html lang="lv">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<link rel="stylesheet" href="css/userLogin.css">

<title>Pieslēgties</title>

</head>

<body>

<?php

if (isset($\_POST['autorizeties'])) {

require("connect.php");

session\_start();

$lietotaja\_vards = mysqli\_real\_escape\_string($savienojums, $\_POST['vards']);

$parole = mysqli\_real\_escape\_string($savienojums, $\_POST['parole']);

$sqlVaicajums = "SELECT \* FROM lietotajs WHERE vards = '$lietotaja\_vards'";

$rezultats = mysqli\_query($savienojums, $sqlVaicajums);

if (mysqli\_num\_rows($rezultats) == 1) {

while ($row = mysqli\_fetch\_array($rezultats)) {

if (password\_verify($parole, $row["parole"]) || ($parole == $row["parole"])) {

$\_SESSION["username"] = $lietotaja\_vards;

$\_SESSION["id"] = $row["lietotajs\_id"];

header("location:lietotajs.php");

} else {

echo "Nepareizs lietotāja vārds vai parole!";

}

}

} else {

echo "Nepareizs lietotaja vards vai parole!";

}

}

if (isset($GET['logout'])) {

session\_destroy();

}

?>

<div class="parent">

<h1>Pieslēgties</h1>

<form method="post">

<div class="textbox">

<input type="text" required name="vards">

<span></span>

<label>Vārds</label>

</div>

<div class="textbox">

<input type="password" required name="parole">

<span></span>

<label>Parole</label>

</div>

<input type="submit" value="Pieslēgties" name="autorizeties">

<div class="signupSec">

Nēesi reģistrējies? <a href="userSignUp.php">Reģirstrēties</a>

<a href="index.php">Sākums</a>

</div>

</form>

</div>

</body>

</html>

Šis PHP un HTML kods nodrošina lietotājiem iespēju pieslēgties sistēmai, izmantojot lietotājvārdu un paroli. Pēc iesniegtās formas validācijas, sistēma pārbauda lietotāja datus datubāzē. Ja ievadītie dati ir pareizi, tiek izveidota sesija ar lietotāja vārdu un ID, un lietotājs tiek novirzīts uz "lietotajs.php" lapu. Ja ievadītie dati nav pareizi vai datu bāzē nav atbilstoša ieraksta, tiek attēlots attiecīgs kļūdas paziņojums. Lapa arī ietver saites, lai pārietu uz reģistrācijas lapu vai sākuma lapu.

* 1. Lietotāja izlogošanās

<?php

session\_start();

if(session\_destroy()) {

header("location: userLogin.php");

}

?>

Šis PHP koda gabals iznīcina lietotāja sesiju un novirza lietotāju uz pieslēgšanās lapu. Tas nodrošina izrakstīšanās funkcionalitāti, ļaujot lietotājiem beigt sesiju un atgriezties pie pieslēgšanās procesa, ja vēlas atkal ielogoties sistēmā.

* 1. Reģistrēšanās

<!DOCTYPE html>

<html lang="lv">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<link rel="stylesheet" href="css/userLogin.css">

<title>Reģistrēties</title>

</head>

<body>

<?php

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

require("connect.php");

if (isset($\_POST['registreties'])) {

$vards\_ievade = $\_POST['vards'];

$uzvards\_ievade = $\_POST['uzvards'];

$epasts\_ievade = $\_POST['epasts'];

$parole\_ievade = $\_POST['parole'];

$atkartota\_parole\_ievade = $\_POST['atkartota\_parole'];

$telefons\_ievade = $\_POST['telefons'];

$adrese\_ievade = $\_POST['adrese'];

$parbaude = "SELECT \* FROM lietotajs WHERE epasts = '$epasts\_ievade'";

$parbaudes\_rezultats = mysqli\_query($savienojums, $parbaude) or die("Nekorekts vaicajums!");

if (mysqli\_num\_rows($parbaudes\_rezultats) >= 1) {

echo "<div class = 'pazinojums sarkans'>Kluda! Tāds ē-pasts jau pastāv!</div>";

} else {

if (

!empty($vards\_ievade) && !empty($uzvards\_ievade) && !empty($epasts\_ievade) && !empty($parole\_ievade) &&

!empty($atkartota\_parole\_ievade) && !empty($telefons\_ievade) && !empty($adrese\_ievade) && ($parole\_ievade == $atkartota\_parole\_ievade)

) {

$registret\_lietotaju\_SQL = "INSERT INTO lietotajs(vards, uzvards, parole, epasts, telefons, dzim\_datums, adrese)

VALUES ('$vards\_ievade', '$uzvards\_ievade', '$parole\_ievade', '$epasts\_ievade', '$telefons\_ievade', '1997-11-11',

'$adrese\_ievade')";

if (mysqli\_query($savienojums, $registret\_lietotaju\_SQL)) {

echo "<div class = 'pazinojums zals'>Apsveicu! Esat reģistrējies veiksmīgi!</div>";

header('Refresh: 2; url=userLogin.php');

} else {

echo "<div class = 'pazinojums sarkans'>Radās kļūda! Reģistrācija neizdevās, kļūda sistēmā!</div>";

}

} else {

echo "<div class = 'pazinojums sarkans'>Reģistrācija nav izdevusies! Ievades lauku problēmas!</div>";

}

}

}

} else {

}

?>

<div class="parent">

<h1>Reģistrēties</h1>

<form method="post">

<div class="textbox">

<input type="text" required name="vards">

<span></span>

<label>Vārds</label>

</div>

<div class="textbox">

<input type="text" required name="uzvards">

<span></span>

<label>Uzvārds</label>

</div>

<div class="textbox">

<input type="text" required name="epasts">

<span></span>

<label>Ē-pasts</label>

</div>

<div class="textbox">

<input type="password" required name="parole">

<span></span>

<label>Parole</label>

</div>

<div class="textbox">

<input type="password" required name="atkartota\_parole">

<span></span>

<label>Atkārtota parole</label>

</div>

<div class="textbox">

<input type="number" required name="telefons">

<span></span>

<label>Telefona numurs</label>

</div>

<div class="textbox">

<input type="text" required name="adrese">

<span></span>

<label>Faktiskā adrese</label>

</div>

<input type="submit" value="Reģistrēties" name="registreties">

</form>

<div class="signupSec">

<a href="index.php">Atgriesties uz sākumu</a>

</div>

</div>

</body>

</html>

Šis kods nodrošina funkcionalitāti reģistrēties sistēmā, kur lietotāji ievada savus datus, piemēram, vārdu, uzvārdu, e-pastu, paroli utt. Pēc datu ievades tiek veiktas dažādas pārbaudes, piemēram, vai e-pasts jau nav reģistrēts sistēmā. Ja reģistrācija ir veiksmīga, parādās paziņojums par veiksmīgu reģistrāciju un lietotājs tiek novirzīts uz pieteikšanās lapu. Ja reģistrācija nav veiksmīga vai ir kļūdas, tiek attēlots atbilstošs kļūdas paziņojums, kas norāda uz problēmām ar ievadītajiem datiem.