



Liepājas Valsts tehnikums

## **Problēmpietiekumu sistēmas izstrāde**

Kvalifikācijas eksāmena dokumentācija

Profesionālā kvalifikācija

Programmēšanas tehniks

Grupas nosaukums

4PT

Projekta izstrādātājs Daniels Vidopskis

Eksāmena datums 2023. gada 21.jūnijs

Liepāja 2023

# Satura radītājs

Satura radītājs	2
Ievads	3
1. Uzdevuma formulējums	4
2. Programmatūras prasību specifikācija	5
2.1. Produkta perspektīva	5
2.2. Sistēmas funkcionālās prasības	5
2.2.1. Apmeklē mājaslapu	5
2.2.2. Microsoft ielogošanās	6
2.2.3. Sākumlapas	7
2.2.4. Pievienot pieteikumu	8
2.2.5. Pieteikuma piešķiršana	9
2.2.6. Pieteikuma atcelšana	9
2.2.7. Saimniecības nodaļas darbinieks	10
2.2.8. It nodaļas darbinieks	10
2.2.9. Pieteikuma rediģēšana un izdzēšana Admin panelī	11
2.2.10. Izrakstīšanās no sistēmas	12
2.3. Sistēmas nefunkcionālās prasības	13
2.3.1. Valoda	13
2.3.2. Saderība ar pārlūkprogrammām	13
2.3.3. Drošība	13
2.3.4. Saderība	13
2.1. Gala lietotāja raksturiežmes	14
3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums	15
3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts	15
3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts	16
4. Sistēmas modelēšana un projektēšana	17
4.1. Sistēmas struktūras modelis	17
4.2. Klašu diagramma / ER diagramma	18
4.3. Funkcionālais un dinamiskais sistēmas modelis	19
4.4. Aktivitāšu diagramma (Activity)	20
4.5. Lietojumgadījumu diagramma (Use Case)	21
4.6. Sistēmas moduļu apraksts un algoritmu shēmas	22
5. Lietotāju ceļvedis	23
6. Testēšanas dokumentācija	26
6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums	26
6.2. Testpiemēru kopa	27
6.3. Testpiemēru žurnāls	33
7. Individuālais ieguldījums	35
8. Secinājumi	36
9. Lietoto saīsinājumu saraksts	37
10. Literatūras un informācijas avotu saraksts	38

## Ievads

Prakses pienākumus es veicu Liepājas Valsts tehnikumā. Tā kā šī mācību iestāde atrodas divās atsevišķās ēkās - Ventspils ielā un Vānes ielā, tad bieži mēdz būt problēmas ar informācijas apmaiņu starp skolas ēkām. Īpaši tad, kad jāpiesaka dažādas tenniskas problēmas un to risinājumi. Šo problēmu atrisināt varētu izstrādāta problēmpieteikumu sistēma, kurā vienkārši un operatīvi būtu iespējams pieteikt problēmu, kā arī sekot līdzi problēmas risinājuma statusam. Tāpēc tika strādāts pie šādas sistēmas izveides.

Tehniskā dokumentācija ir viens no svarīgākajiem dokumentiem kāda projekta izstrādes procesā. Šie dokumenti ietver sevī arī informāciju par pašu "LVT Ticket Support" sistēmu. Problēmpieteikumu atbalsta sistēma ir vietne, kas nodrošina atbalstu skolas pieteikuma problēmu risināšanā LVT skolotājiem. Vietnes mērķis ir palīdzēt skolotājiem un darbiniekiem atrisināt jebkādas problēmas, kas saistītas ar problēmu pieteikumiem, piemēram, neizdevušo savienojumu ar internetu vai salauztu, sabojātu tehniku. Šī iemesla dēļ būtu svarīgi, lai ir izvēle, kurai nodaļai pieteikums paredzēts - IT nodaļai vai saimnieciskajam dienestam. Vēl būtu nozīmīgi, lai problēmas pieteicējs varētu reizē ar problēmas pieteikumu sniegt arī savu komentāru, tas atvieglotu problēmas risinājumu. Sistēma izstrādāta tā, lai problēmas risinātājs redz problēmas atrisinājumu.

## 1. Uzdevuma formulējums

“LVT Ticket Support” ir mājas lapa, kura atrodas uz apache2 servera, tās galvenā funkcija ir izveidot pieteikumu, kurā var ievadīt atrašanās vietu (ielu), telpu, problēmu, piezīmi, komentārus, kā arī izvēli, kurai nodaļai adresēts pieteikums - IT/Saimniecības nodaļu. Pēc tam šī informācija tiek rādīta noteiktai nodaļai, kurā varēs atzīmēt, vai konkrētais pieteikums ir izpildīts vai vēl ir procesā. Mājas lapas administrācija varēs redzēt visus pieteikumus, kā arī to, cik daudz pieteikumu ir izpildīti.

Mājaslapa parāda skolas darbiniekiem, vai viņu pieteikums tiek izpildīts, kā arī tajā var pievienot savu pieteikumu. Mājaslapā pieteikumus var pieteikt gan IT, gan saimniecības nodaļai, kurā var norādīt problēmu, ielu, telpu un piezīmi par risināmo problēmu. Arī steidzamos gadījumos šeit var veikt atzīmi par to, kas atgadījies un vai nepieciešams steidzams risinājums.

Skolotājiem jābūt savam skatam, tas nav saistīts ar IT un saimniecības nodaļas darbiniekiem, un viņi nevar redzēt citu skatu, kas viņiem nav piesaistīts.

Projetam jāizmanto skolas krāsas (zils, balts, sarkani oranžs) jāizveido mājas lapa, kura ir viegli pārskatāma un saprotama, kā arī jābūt trim dažādām lomām (Skolotājs, Administrators, Saimniecības nodaļas darbinieks). Viss projekts ir centrēts uz mācību iestādi Liepājas Valsts tehnikums.

## **2. Programmatūras prasību specifikācija**

### **2.1. Produkta perspektīva**

Produkta perspektīva ir dot iespēju skolas darbiniekiem viegli un ātri pieteikt savu problēmu. Kad sistēma ir veiksmīgi uzstādīta, tad skolas administrācija, kas pārskata visu sistēmu, var deleģēt, kuriem ir piekļuve administrācijas panelim, kur var pārvaldīt un pārskatīt visu sistēmu. Skolas skolotāji var izveidot ātru un vieglu pieteikumu par notikušo problēmu un redzēt, kad problēma ir atrisināta.

### **2.2. Sistēmas funkcionālās prasības**

Sistēma jeb mājaslapa tiek iedalīta trīs apmeklētāju tipos - skolotājs, administrators, darbinieks. Skolotājs ir skolas skolotājs, kurš var izveidot savu pieteikumu par informāciju, kur un kas notika, bet darbinieks ir samniecības nodaļas darbinieks, kurš izpildīs skolotāja izveidoto pieteikumu, kuru varēs atzīmēt kā izpildītu vai arī procesā esošu, bet administrators ir sistēmas pārvaldnieks, kurš tiek klāt pie visām sistēmas funkcijām un visu pārskata.

#### **2.2.1. Apmeklē mājaslapu**

##### Mērķis:

- Mājaslapas adreses ievadīšana jebkurā interneta pārlūkprogrammā.

##### Apstrāde:

- Tiek pārbaudīta, vai ir pareizi uzstādīta sistēma uz serveriem.

##### Izvades dati:

- a) Tiek veiksmīgi ielādēta mājaslapas ielogošanās lapa, kurā tiek prasīts skolas e -pasts un parole.
- b) Kļūdas paziņojums, ka vietne nav sasniedzama.

### 2.2.2. Microsoft ielogošanās

1. tabula

Microsoft prasības

Nosaukums	Obligāts	Piezīme
Skolas e-pasts	Jā	Ieeja tikai ar skolotāju e-pastiem, kuriem nav “sk” pēc @ simbola
Parole	Jā	

Mērķis:

- Neļaut ieeiet mājaslapā skolēniem un citu skolu pedagogiem.

Ievades dati:

- a) Ierakstot skolas e-pastu “email” laukumā un paroli “password” laukumā.

Apstrāde:

- Pēc pogas “login” uzspiešanas dati tiek aizsūtīti uz “Microsoft”, kur tie tiek pārbaudīti un verificēti, vai patiešām ir pareizi ievadīti.

Izvades dati:

- a) Pēc veiksmīgas datu aizsūtīšanas un verificācijas aizved lietotāju uz paredzēto sāklapību.
- b) Pēc neveiksmīgas datu aizsūtīšanas vai verificācijas parāda lietotājam, vai parole vai e-pasts ir pareizi ievadīts.

### 2.2.3. Sākumlapas

#### Mērķis:

- Pēc ieiešanas mājaslapā redzama sākumlapa, kurā redzams skolotāja izveidotais pieteikums/ pieteikumi, kuru pie vajadzības var arī atcelt.

#### Ievades dati:

-

#### Apstrāde:

- Saņem pieteikuma datus no datubāzes, kuri ir veidoti no tā lietotāja datiem, kurš ir iegājis mājaslapā, un tie tiek sakārtoti no jaunākā uz vecāko pieteikumu.

#### Izvades dati:

- a) Tiek veiksmīgi saņemti dati no datubāzes, parāda pieteikumus no jaunākā uz vecāko.
- b) Tabula ir tukša, ja nav izveidots neviens pieteikums.

## 2.2.4. Pievienot pieteikumu

2. tabula

Pieteikuma prasības

Nosaukums	Obligāts	Piezīme
Iela	Jā	
Telpa	Jā	
Nodaļa	Jā	
Problēma	Jā	
Piezīme	Nē	Piezīme nav obligāta jābūt izpildītai lai pievienot pieteikumu

### Mērķis:

- Pievienot skolotāja pieteikumu/ piteikumus datubāzē, kuru varēs redzēt gan dabinieki, gan administrācija, gan skolotājs. Būs redzama informācija, kurā skolas ēkā, kurā telpā problēma risināma, kā arī tas, vai pieteikums ir adresēts IT vai saimniecības nodaļai.

### Ievades dati:

- Izvēlēties Vānes ielas vai Ventspils ielas ēku, norādīt telpu, izvēlēties, vai IT, vai saimniecības nodaļu, ierakstīt problēmu un ierakstīt piezīmi (tā nav obligāta).

### Apstrāde:

- Saņem pieteikuma datus no datubāzes, kuri ir veidoti no tā lietotāja, kurš iegājis mājaslapā, un sakārto no jaunākā uz vecāko pieteikumu.

### Izvades dati:

1. Aizved uz sākulapu, kurā būs redzami vecie izveidotie problēmpieteikumi un jaunākie, kuri tikko tika izveidoti.
2. Parāda kļūdu, kurā ir teikts, ka viens vai vairāki obligātie lauki nav izpildīti.



### **2.2.5. Pieteikuma piešķiršana**

Mērķis:

- Sistēmai automātiski ir jāpiešķir iesniegtās biļetes atbilstošam personālam vai nodaļām, lai zinātu, kuram tas pieder.

Ievades dati:

- Izvēlēties Vānes ielas vai Ventspils ielas ēku, norādīt telpu, izvēlēties, vai IT, vai saimniecības nodaļu, ierakstīt problēmu un ierakstīt piezīmi (tā nav obligāta).

Apstrāde:

- Pieteikumam pievieno unikālu atslēgu pie lietotāja e- pasta.

Izvades dati:

- Nav izvades dati, viss notiek mājaslapas aizmugurē jeb backend.

### **2.2.6. Pieteikuma atcelšana**

Mērķis:

- Skolotājs var atcelt pieteikuma gadījumā, ja ir ievadīta nepareiza informācija vai problēma tiek atrisināta pati no sevis.

Ievades dati:

.

Apstrāde:

Tiek atzīmēts, ka pieteikums ir atcelts un saglabājas datubāzēs.

Izvades dati:

Noņem pieteikumu no sākulapas.

## **2.2.7. Saimniecības nodaļas darbinieks**

### Mērķis:

Saimniecības nodaļas darbiniekiem sava sākulapa, lai būtu vieglāk pārskatīt pieteikumus un problēmas, kas ir noticis. Kā arī, lai telefonā būtu viegli izlasāms.

### Ievades dati:

Skolas e-pasts

### Apstrāde:

Tiek salīdzināts, vai e-pasts atbilst saimniecības nodaļas e pastiem un sāk saimniecības nodaļas sākulapas ielādi.

### Izvades dati:

- a) Saimniecības nodaļas darbinieka sākulapa.
- b) Skolotāja pieteikumi, kurā ir norādīta Telpa, Iela, Problēma un pieteikums.

## **2.2.8. It nodaļas darbinieks**

### Mērķis:

- IT nodaļas darbiniekiem ir administrācijas panelis, kur var pārvaldīt visu sistēmu, kā arī pievienotos pieteikumus.

### Ievades dati:

- Skolas e-pasts.

### Apstrāde:

- Tiek salīdzināts vai e-pasts atbilst ar IT nodaļas e-pastiem un sāk administrācijas paneļa sākulapas ielādi

### Izvades dati:

- a) Admin paneļa sākulapa.
- b) Skolotāja pieteikumi, kurā ir norādīta Telpa, Iela, Problēma un pieteikums.
- c) Rāda pieteikuma statistiku "Kopā", "Pabeigti", "Procesā", "Nav pabeigti".

## 2.2.9. Pieteikuma rediģēšana un izdzēšana Admin panelī

3. tabula

Pieteikuma rediģēšana un izdzēšana

Nosaukums	Obligāts	Piezīme
Iela	Nē	
Telpa	Nē	
Nodaļa	Nē	
Problēma	Nē	
Piezīme	Nē	

### Mērķis:

Rediģēt pieteikuma gadījumā, vai ir izdarīts, vai ir nepareiza informācija ielikta pieteikumā, kā arī dzēst pielikumu.

### Ievades dati:

Izvēlēties pieteikumu no saraksta, kuru vēlas rediģēt vai dzēst.

### Rediģēšana:

Izvēloties rediģēšanas iespēju, lietotājam tiek atvērta forma ar pašreizējiem pieteikuma datiem.

Izmantojot šo formu, administrators var labot pieteikuma datus, piemēram, izvēlēties citu ēku, telpu, nodaļu, problēmu vai piezīmi.

### Dzēšana:

Izvēloties dzēšanas iespēju, administratoram tiek parādīts apstiprinājuma dialogs, lai pārliecinātos par pieteikuma dzēšanas nodomu. Apstiprinot dzēšanu, pieteikums tiek dzēsts no sistēmas.

### Izvades dati:

Pēc veiktajām izmaiņām vai dzēšanas, pieteikumu saraksts tiek atjaunots, parādot jaunāko informāciju. Ja rediģēšana vai dzēšana netiek veikta, pieteikumu saraksts paliek nemainīgs.

#### **2.2.10. Izrakstīšanās no sistēmas**

Mērķis:

Izrakstīties no sistēmas, lai kāds cits netiek iekšā ar tavu lietotāju.

Ievades dati:

-

Apstrāde:

Tiek aizsūtīts pieprasījums uz Microsoft, lai izlogojas no Microsoft lietotāja profila un nesaglabā informāciju par profilu.

Izvades dati:

a) Aizved lietotāju uz sistēmas login mājaslapu.

## **2.3. Sistēmas nefunkcionālās prasības**

### **2.3.1. Valoda**

Mājaslapa tika izveidota latviešu valodas lietotājiem.

### **2.3.2. Saderība ar pārlūkprogrammām**

Mājaslapa un administrācijas pārvaldības paneļa sistēmas funkcionalitātei jālieto populārākās pārlūkprogrammas, kā piemēram, “Mozilla Firefox”, “Opera”, “Opera GX”, “Google Chrome”, “Microsoft Edge”.

### **2.3.3. Drošība**

Sistēmai jānodrošina lietotāju datu drošība un konfidencialitāte, tostarp lietotāju pieteikšanās datus un personīgo informāciju. Tā jāizmanto atbilstošas drošības pasākumi, piemēram, datu šifrēšana, droša savienojuma protokols (HTTPS) un piekļuves kontroles, lai pasargātu pret neautorizētu piekļuvi un datu noplūdēm.

### **2.3.4. Saderība**

Ekrāna izmēra pielāgojamība: Sistēmai jābūt pielāgojamai dažādu ekrāna izmēru un izšķirtspēju ierīcēm. Neskatoties uz to, vai lietotājs izmanto lielo datora ekrānu, planšetdatoru vai mobilās ierīces mazāko ekrānu, sistēmai jāsniedz optimāla lietojamība. Visas funkcijas un satura elementi ir jāattēlo pareizi un jābūt viegli sasniedzamiem neatkarīgi no ekrāna lieluma.

Atbalsts dažādiem mobilo telefonu modeļiem: sistēmai jābūt saderīgai ar plašu mobilo telefonu modeļu klāstu, ko izmanto lietotāji. Lai nodrošinātu pilnīgu lietojamību, sistēmai jābūt pielāgojamai dažādiem operētājsistēmu versijām un ekrāna izmēriem, kas tiek izmantoti populārākajos telefona modeļos. Tas ietver gan Android, gan iOS platformas.

## **2.1. Gala lietotāja raksturiezīmes**

### Skolotājs:

Skolotājs ir Liepājas Valsts tehnikuma darbinieks, kurš vēlas izmantot problēmpieteikumu sistēmu, lai viegli un ātri pieteiktu problēmas vai nepilnības, kas saistītas ar skolas iekārtām. Skolotājam ir piekļuve savam skatam, kur viņš var izveidot jaunu pieteikumu, norādot problēmas atrašanās vietu, telpu un pievienot komentārus par problēmu. Skolotājs arī var sekot līdzi pieteikuma statusam un redzēt, vai problēma ir izpildīta vai vēl tiek risināta. Viņam ir svarīgi, lai sistēma būtu viegli saprotama un ērti lietojama.

### Administrators:

Administrators ir sistēmas pārvaldnieks, kuram ir pilna piekļuve problēmpieteikumu sistēmas funkcijām. Viņš var pārvaldīt un pārskatīt visus pieteikumus, kā arī piešķirt un atzīmēt tos kā izpildītus. Administrators ir atbildīgs par sistēmas uzturēšanu un darbības nodrošināšanu. Viņš var arī veikt citus administratīvus uzdevumus, piemēram, pārvaldīt lietotāju kontus vai piekļuves tiesības. Administrators novērtē sistēmas darbību no globālās perspektīvas un nodrošina tās efektīvu darbību.

### Darbinieks:

Darbinieks ir Liepājas Valsts tehnikuma samniecības nodaļas pārstāvis, kurš atbildīgs par pieteikumu izpildi un problēmu risināšanu. Viņš iegūst informāciju par izveidotiem pieteikumiem, kuras ir radījuši skolotāji, un var atzīmēt, vai pieteikums ir izpildīts vai vēl tiek risināts. Darbinieks strādā, lai nodrošinātu, ka skolas iekārtas un infrastruktūra ir darbības kārtībā un ka problēmas tiek atrisinātas efektīvi un laicīgi. Viņam ir nepieciešama viegli lietojama un pārskatāma problēmpieteikumu sistēma, kas ļauj viņam veikt savus darbības ātri un efektīvi.

### **3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums**

#### **3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts**

##### HTML:

Ir dažas alternatīvas valodas HTML, piemēram, Markdown, YAML un XML.

Markdown ir viegla un lasāma marķēšanas valoda, kas paredzēta teksta formatēšanai tīmekļa vietnēs un dokumentos. Tā piedāvā vienkāršas sintakses konstrukcijas, kas ļauj veidot formatētu tekstu bez sarežģītām HTML etiķetēm. Markdown ir populārs starp blogeriem, programmētājiem un citiem, kas vēlas viegli formatēt tekstu, bet nevēlas lietot HTML.

##### PHP:

Es izvēlējos izmantot PHP vietā citām alternatīvām, jo tā bija vispiemērotākā valoda šim konkrētajam projektam.

Pieredze: man pašam ir neliela pieredze ar PHP valodas pielietojumu. Plaša atbalsta kopiena: PHP ir viena no populārākajām serverpusēm izpildāmajām valodām, un tā ir plaši izmantota visā pasaulē. Tas nozīmē, ka ir pieejami lieliski resursi, kas atvieglo izstrādi un problēmu risināšanu.

Integrācija ar datu bāzēm: PHP piedāvā lielisku integrāciju ar dažādām datu bāzēm, piemēram, MySQL un PostgreSQL. PHP ir platformas neatkarīga valoda, kas nozīmē, ka es varu to izmantot dažādos operētājsistēmās un serveru vidēs, piemēram, Linux, Windows vai macOS, bez ierobežojumiem.

##### CSS:

CSS preprocesori ir valodas, kas izmanto priekšprocesoru pielietošanai CSS stilu lapām. Tie ļauj rakstīt CSS kodu, kas ir atjaunināms, atkārtoti izmantojams un strukturizēts veids. Preprocesori piedāvā vairākas priekšrocības.

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

Alternatīvas valodas CSS:

Sass (Syntactically Awesome Style Sheets): Sass ir CSS priekšprocesors, kas paplašina CSS iespējas, pievienojot maināmos, iekļaušanu, mixinus un citus elementus. Tas palīdz optimizēt CSS izstrādes procesu un padara stilu lapas vieglāk uzturamas.

Less (Leaner Style Sheets): Less ir vēl viens CSS priekšprocesors, kas piedāvā līdzīgas iespējas kā Sass, piemēram, maināmos, mixinus un iekļaušanu. Tas cenšas vienkāršot CSS izstrādi un uzlabot kodu organizāciju.

Alternatīvas valodas HTML:

Markdown: Markdown ir viegla marķēšanas valoda, kas ļauj rakstīt formatētu tekstu, izmantojot vienkāršu sintaksi. To plaši izmanto dokumentācijas, readme failu un vienkāršu tīmekļa lapu izveidē.

Pug (agrāk pazīstams kā Jade): Pug ir veidnes valoda, kas vienkāršo HTML izveidi, izmantojot atkāpšanu un saīsinātu sintaksi. Tā piedāvā iespējas kā mainīgos, nosacījumus, ciklus un iekļaušanas funkciju, padarot dinamisko HTML saturu vieglāk ģenerējamu.

Alternatīvas valodas JavaScript:

TypeScript ir JavaScript papildinājums, kas pievieno statisko tipizēšanu un citas uzlabotas funkcijas, lai uzlabotu izstrādes pieredzi. Uzlabo kodu uzturamību un nodrošina labāku rīku atbalstu.

Alternatīvas valodas PHP:

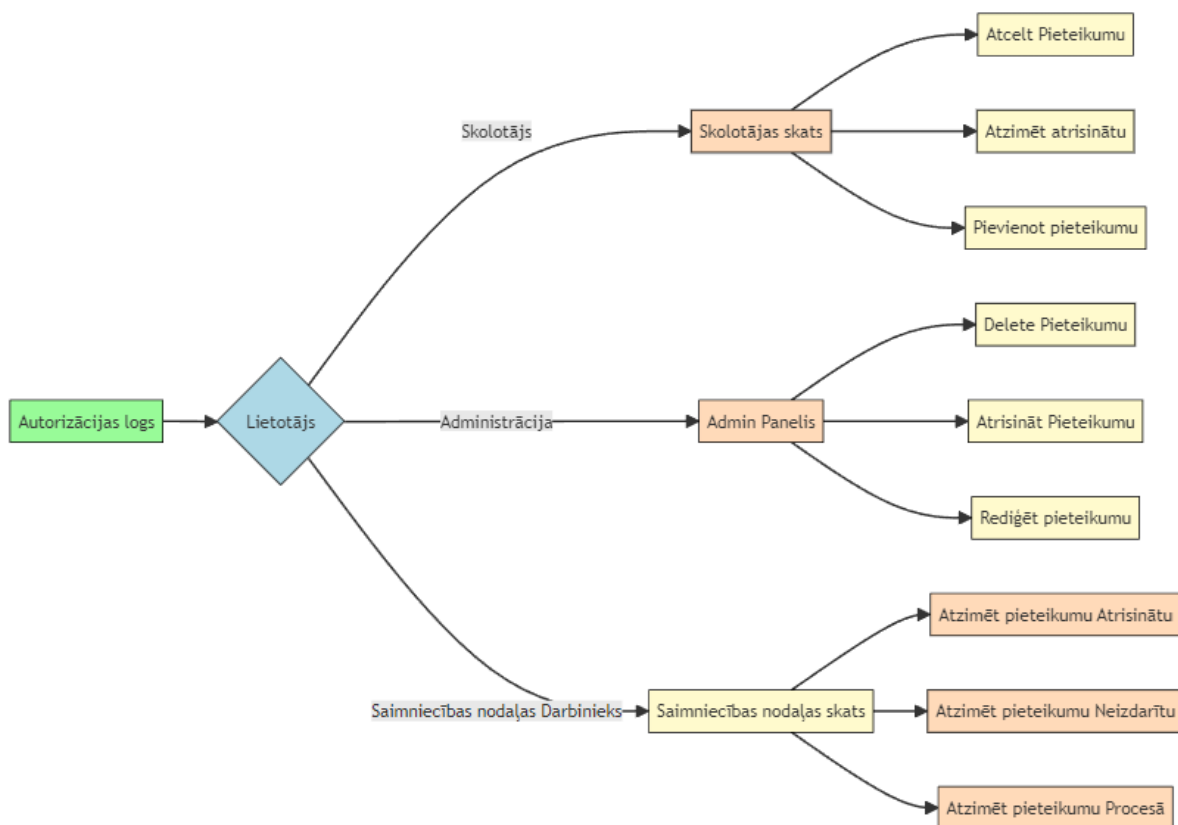
Python: Python ir daudzpusīga valoda ar plašu pielietojumu, tostarp tīmekļa izstrādē. Tai ir ietvaru sistēmas kā Django un Flask, kas nodrošina spēcīgus rīkus tīmekļa lietojumprogrammu izveidei ar tīru un uzturamām kodu.



## 4. Sistēmas modelēšana un projektēšana

### 4.1. Sistēmas struktūras modelis

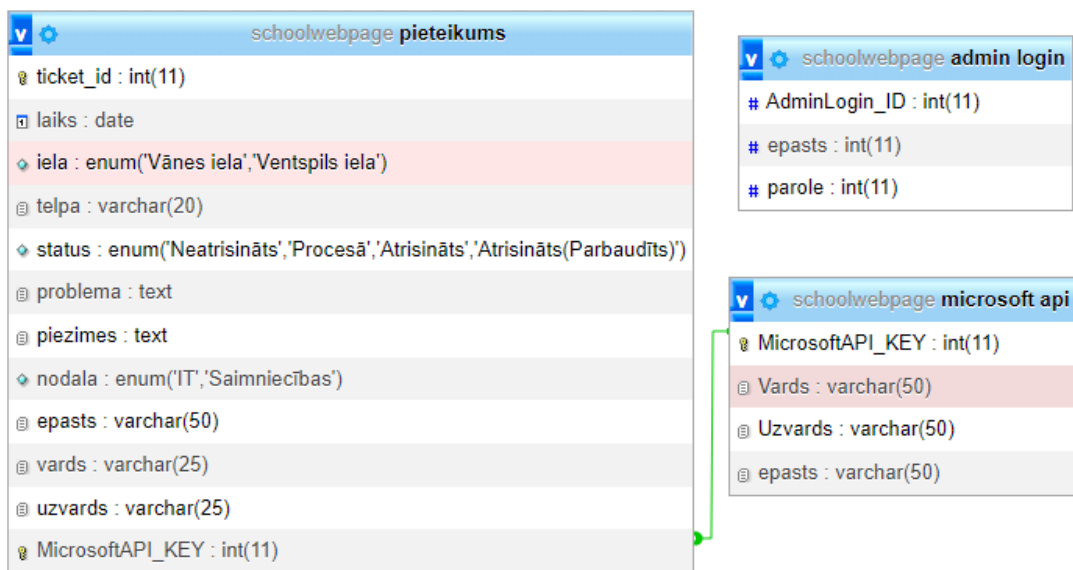
Šī Sistēmas struktūras modelis diagramma vizuāli attēlo, kā lietotāji izmanto sistēmu un kādas darbības viņi var veikt, atkarībā no savas lomas. Tas palīdz skaidri parādīt sistēmas procesu un uzvedību, kas ir noderīgi, lai saprastu lietotāju darbību gaitu un iegūtu vispārēju priekšstatu par sistēmas funkcionalitāti.



1.attēls. struktūras attēls

## 4.2. Klašu diagramma / ER diagramma

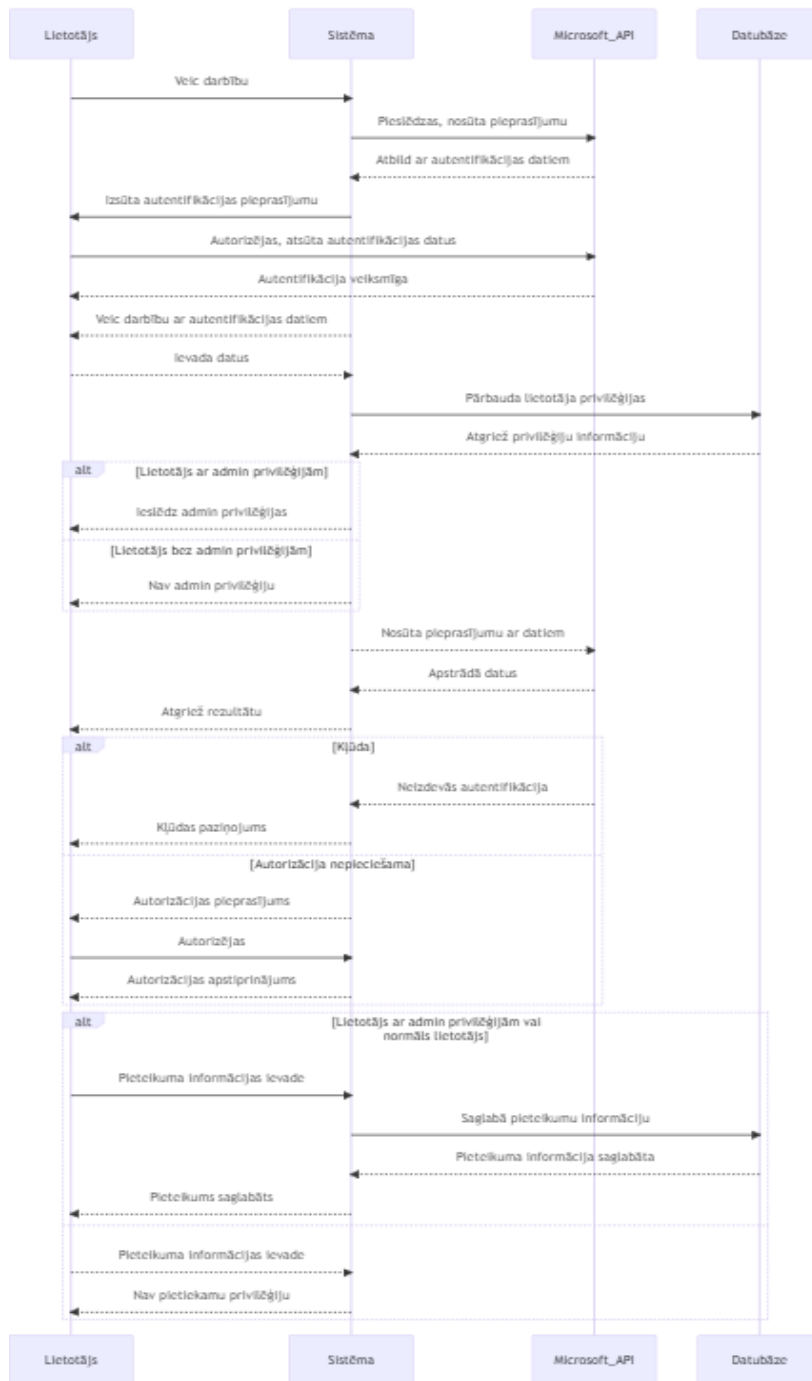
Klašu diagrammā var redzēt, ka tiek izmantots Microsoft API. Ar Microsoft API palīdzību ir iespējams iegūt lietotāja epastu, vārdu un uzvārdu (Skat. 2. attēlu).



2.attēls. Klašu diagram attēls

### 4.3. Funkcionālais un dinamiskais sistēmas modelis

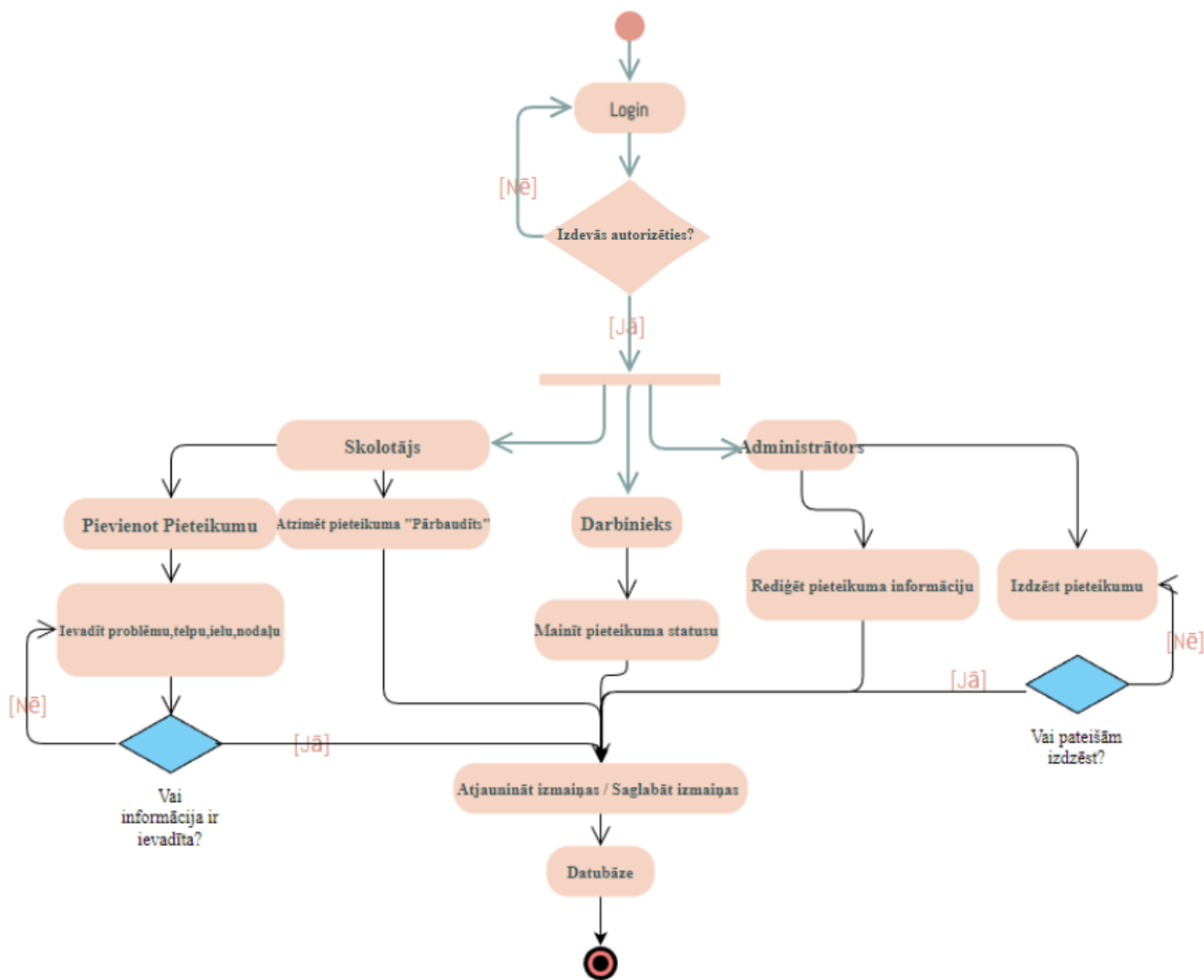
Kopumā Funkcionālais un dinamiskais sistēmas modelis nodrošina rīkus un metodes, lai vizualizētu un izprastu sistēmas struktūru, funkcijas un uzvedību.



3.attēls. Dinamiskais sistēmas modeļa attēls

#### 4.4. Aktivitāšu diagramma (Activity)

Darbību diagramma attēlo darbību plūsmu ienākšanu sistēmā. Tā demonstrē ienākšanas procesu un funkcionalitātes, kas ir pieejamas dažādām lietotāju lomām (Skat. 4. attēlu).

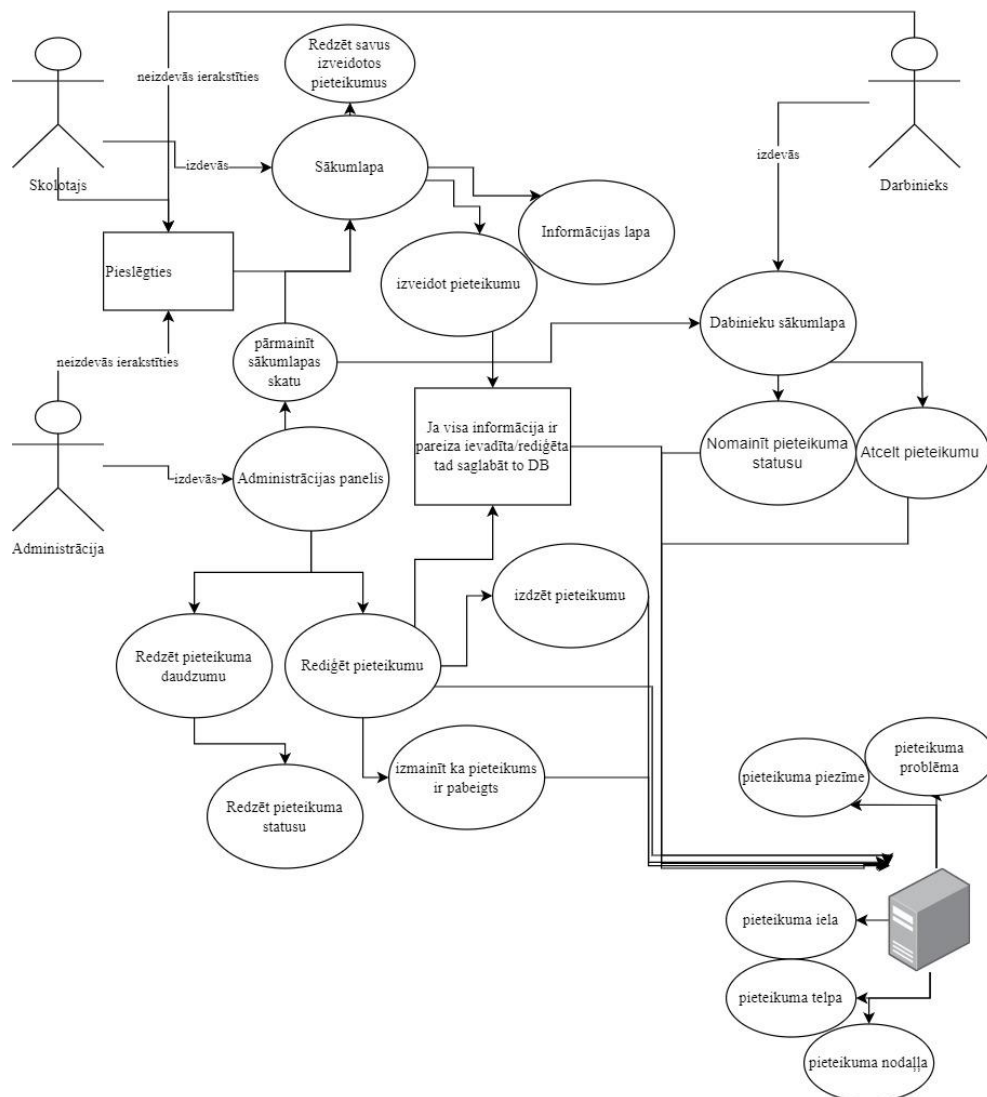


4.attēls. Aktivitāšu diagrammas attēls

#### 4.5. Lietojumgadījumu diagramma (Use Case)

Diagrammā attēlots sistēmas lietotāju mijiedarbība, ieskicējot funkcijas un iespējas, ko piedāvā sistēma.

Lietojumgadījumu diagrammā ir iekļauti trīs galvenie faktori: "Skolotājs", "Administrācija" un "Darbinieks" (Skat. 5. attēlu).



5.attēls. Lietojumgadījumu diagrammas attēls

#### 4.6. Sistēmas moduļu apraksts un algoritmu shēmas

Sistēmas moduļa apraksts:

Sistēmas modulis ir daļa no lielākas skolas tīmekļa lapas lietotnes, kas ļauj skolēniem un skolotājiem mijiedarboties un pārvaldīt dažādus skolas aspektus, piemēram, iesniegt problēmu ziņojumus IT vai administratīvām problēmām un pārvaldīt pieteikuma informāciju. Sistēmas modulis ir atbildīgs par šo funkcionalitāšu aizmugurējās daļas loģiku un datu pārvaldību.

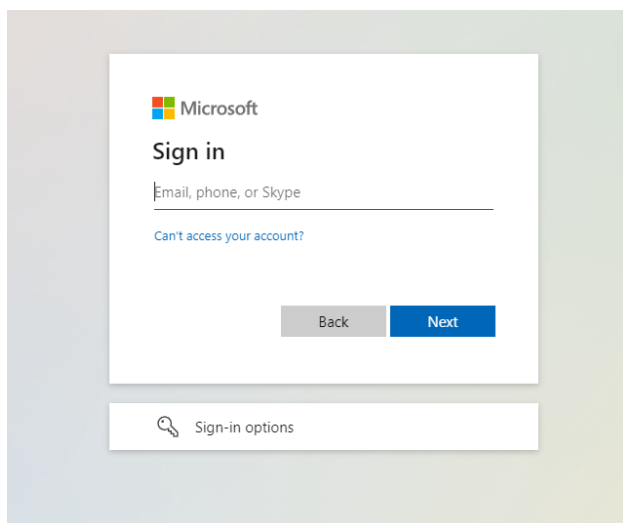
Algoritma shēma:

1. Inicializē sistēmas moduli, pieslēdzoties skolas tīmekļa lapas datu bāzei un iestatot nepieciešamos SQL režīmus un laika joslu.
2. Definē datu struktūras datu bāzes tabulām, piemēram, 'apstiprinājums', 'ieteikums', 'problēma' un 'skolēns'. Šīs datu struktūras ietver kolonnas, datu tipus un ierobežojumus katrai tabulai.
3. Īsteno funkcijas, kas apstrādā CRUD (izveidot, lasīt, atjaunināt, dzēst) darbības katrai tabulai, piemēram, ievietot jaunus ierakstus, izgūt ierakstus, atjaunināt ierakstus un dzēst ierakstus, izmantojot SQL vaicājumus.
4. Īsteno funkcijas, kas apstrādā konkrētas funkcionalitātes, piemēram, apstrādā studentu konsultāciju pieprasījumus, apstrādā studentu iesniegtos problēmu ziņojumus, pārvalda apstiprinājumus konsultāciju pieprasījumiem un atjaunina problēmu ziņojumu statusus.
5. Īsteno funkcijas, kas nodrošina datu validāciju un kļūdu apstrādi, piemēram, pārbauda derīgus ievades vērtības, apstrādā datu bāzes kļūdas un nodrošina atbilstošus kļūdu ziņojumus lietotājiem.
6. Īsteno funkcijas, kas nodrošina biznesa loģiku, piemēram, pārbauda konfliktus konsultāciju pieprasījumos, aprēķina datumus konsultācijām un pār????????
7. Sistēmā tiek izmantots Microsoft Azure API, kas ļauj dabūt informāciju par lietotāju profilu. Piemēram, vai lietotājs ir skolotājs vai nē. Mācību iestādes lietotāji tiek sistēmā ar Microsoft Azure API palīdzību.

## 5. Lietotāju ceļvedis

Lietošanas instrukcijas sniedz informāciju par to, kā izmantot problēmpieteikumu sistēmu. Šis ceļvedis ir paredzēts visiem sistēmas lietotājiem, tostarp skolotājiem, administrācijas un IT/saimniecības nodaļas darbiniekiem. Zemāk ir norādīti soļi un funkcijas, lai jūs varētu efektīvi izmantot Problēmpieteikumu sistēmu.

### 1. Pieteikšanās sistēmā:



### 6.attēls. mājaslapas attēls

- Atveriet jebkuru interneta pārlūkprogrammu un ievadiet Problēmpieteikumu sistēmu adresi.
- Ievadiet savu skolas e-pasta adresi un paroli.

### 2. Pieteikumu izveide:



### 7.attēls. mājaslapas attēls

- Pēc pieteikšanās sistēmā noklikšķiniet uz "Pievienot pieteikumu".
- Izvēlieties skolas ēku, kurā ir problēma (Vānes iela vai Ventspils iela).
- Norādiet telpu, kurā problēma notiek.
- Izvēlieties, vai pieteikums ir adresēts IT vai saimniecības nodaļai

- Ierakstiet problēmas aprakstu un piezīmes par risināmo problēmu.

#### 8.attēls. mājaslapas attēls

- Noklikšķiniet uz "Pievienot" pogas, lai iesniegtu pieteikumu.

#### 3. Skatīt un pārvaldīt pieteikumus:

Iela un telpa/datums	Problēma	Piezīme	Statuss	IT/Saimniecības nodaļa	Apstiprināt
Vānes iela: A203 2023-06-12	Man pazuda siena		Neatrisināts	IT nodaļa	Atcelt

#### 9.attēls. mājaslapas attēls

- Pēc pieteikuma izveides jūs varēsiet redzēt savu pieteikumu sākumlapā.
- Pieteikumi tiks sakārtoti no jaunākā uz vecāko.
- Skolotāji varēs redzēt tikai savus izveidotos pieteikumus un to statusus.

#### 4. Apstiprināt pieteikumu

- Pēc pieteikuma izveides jūs varēsiet redzēt savu pieteikumu sākumlapā.
- Ja pieteikums ir atrisināts, jūs varēsiet atzīmēt ka ir pārbaudīts.

Atrisināts	IT nodaļa	Izdarīts	Neizdarīts
------------	-----------	----------	------------

#### 10.attēls. mājaslapas attēls

#### 5. Atcelt pieteikumu:

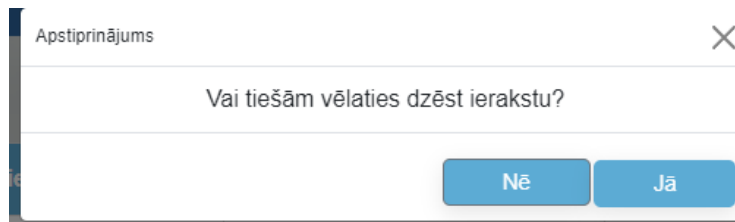
- Pēc pieteikuma izveides jūs varēsiet redzēt savu pieteikumu sākumlapā.
- Ja jūs esat izveidojuši pieteikumu ar nepareizu informāciju, jūs varat vinu atcelt.

Statuss	IT/Saimniecības nodaļa	Apstiprināt
Neatrisināts	IT nodaļa	Atcelt

#### 11.attēls. mājaslapas attēls



- Pārbaudiet vai jūs patiešām gribat atcelt pieteikumu, ja ir pārbaudīts tad uzpiežat uz “Jā” pogas.



**12.attēls. mājaslapas attēls**

6. Iziēt no sistēmas:

- Lai izietu no "Problēmpieteikuma" sistēmas, noklikšķiniet uz savu vārdu un uzvārdu.
- Pēc tam noklikšķiniet uz “izrakstīties” pogu.
- Lūdzu neaizmirstiet iziet no sistēmas, ja lietojat publisku datoru.

## **6. Testēšanas dokumentācija**

### **6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums**

Melnā kaste (Black Box) testēšana ir piemērota izvēle, ja nepieciešams pārbaudīt tīmekļa vietnes funkcionalitāti un lietotāja interfeisu, nemazinoties ar iekšējo kodu vai sistēmas struktūru. Pamatojumi par šīs metodes izvēli ietver:

Lietotāja perspektīva: Melnā kaste (Black Box) testēšana ļauj pārbaudīt tīmekļa vietni tā, kā to darītu galīgais lietotājs, koncentrējoties uz pareizu darbību un interakciju ar lietotāja interfeisu.

Neatkarība no iekšējiem risinājumiem: Ar šo metodi nav nepieciešama detalizēta iekšējo komponentu vai kodu izpratne, tādēļ var veikt testēšanu neatkarīgi no to realizācijas detaļām.

Pārbaude no vairākiem aspektiem: Melnā kaste (Black Box) testēšana ļauj pārbaudīt dažādus aspektus, piemēram, funkcionalitāti, ievades validāciju, datu saglabāšanu un sistēmas atbildes laiku.

## 6.2. Testpiemēru kopa

Prasības ID	Prasība
PR.01	Mājaslapas uzstādīšana
<b>Autorizācijas Logs</b>	
PR.02	Autorizēties sistēmā
PR.03	Microsoft API savienojums
PR.04	Administrācijas skats
PR.05	Skolotāju skats
PR.06	Saimniecības skats
<b>Skolotājas skats</b>	
PR.07	Pieteikuma parādīšana mājaslapā
PR.08	Pieteikuma pievienošana
PR.09	Pieteikuma apstiprināšana
PR.10	Pieteikuma atcelšana
PR.11	Pieteikums nav izdarīts
<b>Administrācijas panelis</b>	
PR.12	Pieteikuma atrisināšana
PR.13	Pieteikuma izdzēšana
PR.14	Pieteikuma rediģēšana
<b>Saimniecības nodaļas darbinieku skats</b>	
PR.15	Pieteikuma parādīšana mājaslapā
PR.16	Pieteikuma statusa maiņa uz "Izdarīts"
PR.17	Pieteikuma statusa maiņa uz "Neizdarīts"
PR.18	Pieteikuma statusa maiņa uz "Iesākts"
PR.19	Atrisināto pieteikumu vēsture

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra izpildes sagaidāmais rezultāts	Prasību identifikators
<b>TP.01</b>	Mājaslapas uzstādīšana		1) Sistēmas faili ielikti servera mapītē 2) Serveris tiek palaists	-	Tiek parādīts mājaslapas autorizācijas logs	PR.01
<b>TP.02</b>	Autorizēties sistēma ar Microsoft API	Jābūt sistēmas mājaslapā	Ierakstīta sistēmas adrese pārlūkprogrammā	Sistēmas adrese		PR.02
<b>TP.03</b>	Savienojums ar Microsoft	Savieno sistēmu ar Microsoft API serveriem	Tiek aizsūtīts pieprasījums Microsoft API	Skolas e-pasts un parole	Tiek pārbaudīti ievades dati Microsoft pusē	PR.03
<b>TP.04</b>	Lietotājs nav autorizēts sistēmā	1) Lietotājs ir iegājis ar savu personālo e-pastu 2) Lietotājs ir skolnieks, kurš nav autorizēts sistēmā		E-pasts, kas nav autorizēts sistēmā	Parāda, ka lietotājs nav autorizēts sistēmā un ka nav mācību iestādes lietotājs	PR.03
<b>TP.05</b>	Administrācijas paneļa skats	Autorizējies mājaslapā ar lietotāju, kuram ir administrācijas piekļuve		Mācību iestādes e-pasts ar administrācijas piekļuvi	Parāda administrācijas paneli	PR.04

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra izpildes sagaidāmais rezultāts	Prasību identifikators
<b>TP.06</b>	Pieteikuma atrisināšana administrācijas panelī	1) Lietotājam jābut administratoram 2) Jābut atvērtam administrācijas panelim	1) Izvēlēties, kurš pieteikums ir atrisināts 2) Uzspiež uz “Atrisināts” pogu		Administrācija var atzīmēt pieteikumu kā atrisinātu datubāzē	PR.12
<b>TP.07</b>	Pieteikuma izdzēšana administrācijas panelī	1) Lietotājam jābut administratoram 2) Jābut atvērtam administrācijas panelim	1) Izvēlēties kuru pieteikumu izdzēst 2) Apstiprināt vai patiešām izdzēst pieteikumu		Administrācija var izdzēst pieteikumu no datubāzes	PR.13
<b>TP.08</b>	Pieteikuma rediģēšana administrācijas panelī	1) Lietotājam jābut administratoram 2) Jābut atvērtam administrācijas panelim	1) Izvēlēties, kuru pieteikumu rediģēt 2) Ierakstīt vai izvēlēties pareizos datus		Administrācija var rediģēt pieteikumu datubāzē	PR.14
<b>TP.09</b>	Skolotāja sākumlapa	Autorizējies mājaslapā ar lietotāju, kuram ir skolotāja piekļuve		Mācību iestādes e-pasts ar skolotāju piekļuvi	Parāda skolotāja sākumlapu	PR.05
<b>TP.10</b>	Skolotāja sākumlapas pieteikumi	Autorizējies mājaslapā ar lietotāju, kuram ir skolotāja piekļuve			Parādīt lietotāja izveidotos pieteikumus sākumlapā	PR.07

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra izpildes sagaidāmais rezultāts	Prasību identifikators
<b>TP.11</b>	Skolotāja pieteikuma pievienošana	Autorizējies mājaslapā ar lietotāju, kuram ir skolotāja piekļuve	1) Uzspiežot uz pieteikuma pievienošanas pogas 2) Izvēlēties ielu 3) Ierakstīt telpas Nr. 4) Izvēlēties nodaļu 5) Ierakstīt pieteikumu 6) Ir iespēja ierakstīt piezīmes	1) Iela 2) Telpa 3) Nodaļa 4) Problēma 5) Piezīme	Tiek pievienots pieteikums datubāzē	PR.08
<b>TP.12</b>	Skolotāja sākumlapas pieteikuma atcelšana	1) Autorizējies mājaslapā ar lietotāju, kuram ir skolotāja piekļuve 2) Ir jau izveidoti pieteikumu	1) Uzspiežot uz "Atcelt" pogas 2) Apstripināt pieteikuma atcelšanu		Tiek atcelts pieteikums datubāzē	PR.10
<b>TP.13</b>	Skolotāja sākumlapas pieteikuma apstiprināšana	1) Autorizējies mājaslapā ar lietotāju, kuram ir skolotāja piekļuve 2) Ir jau izveidoti pieteikumu 3) Pieteikums ir atrisināts	1) Uzspiežot uz "Izdarīts" pogas		Tiek apstripināts pieteikums datubāzē	PR.09
<b>TP.14</b>	Skolotāja sākumlapas pieteikums neizdarīts	1) Autorizējies mājaslapā ar lietotāju, kuram ir skolotāja piekļuve 2) Ir jau izveidoti pieteikumu 3) Pieteikums ir atrisināts	1) Uzspiežot uz "Neizdarīts" pogas		Datubāzē pieteikums tiek atzīmēts kā neizdarīts	PR.11

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra izpildes sagaidāmais rezultāts	Prasību identifikators
<b>TP.15</b>	Saimniecības nodaļas dabnieka sākulapa	Autorizējies mājaslapā ar lietotāju, kuram ir saimniecības nodaļas piekļuve		Mācību iestādes e-pasts ar saimniecības nodaļas piekļuvi	Parādās saimniecības nodaļas sākulapa	PR.06
<b>TP.16</b>	Saimniecības nodaļas dabnieka sākulapas pieteikumi	1) Autorizējies mājaslapā ar lietotāju, kuram ir saimniecības nodaļas piekļuve 2) Skolotāji ir izveidojuši pieteikumus priekš saimniecības nodaļas			Parādīt lietotāja izveidotos pieteikumus sākulapā	PR.15
<b>TP.17</b>	Saimniecības nodaļas dabnieka pieteikums izdarīts	1) Autorizējies mājaslapā ar lietotāju, kuram ir saimniecības nodaļas piekļuve 2) Skolotāji ir izveidojuši pieteikumus priekš saimniecības nodaļas			Datubāzē pieteikums tiek atzīmēts kā izdarīts	PR.16
<b>TP.18</b>	Saimniecības nodaļas dabnieka pieteikums neizdarīts	1) Autorizējies mājaslapā ar lietotāju, kuram ir saimniecības nodaļas piekļuve 2) Skolotāji ir izveidojuši pieteikumus priekš saimniecības nodaļas			Datubāzē pieteikums tiek atzīmēts kā neizdarīts	PR.17

Testpiemēra ID	Testpiemēra nosaukums	Testpiemēra izpildes nosacījumi	Testpiemēra izpildes soļi	Testpiemēra ievades dati	Testpiemēra izpildes sagaidāmais rezultāts	Prasību identifikators
<b>TP.19</b>	Saimniecības nodaļas dabnieka pieteikums procesā	1) Autorizējies mājaslapā ar lietotāju, kuram ir saimniecības nodaļas piekļuve 2) Skolotāji ir izveidojuši pieteikumus priekš saimniecības nodaļas			Datubāzē pieteikums tiek atzīmēts kā procesā	PR.18
<b>TP.20</b>	Saimniecības nodaļas dabnieka atrisināto pieteikumu vēsture	1) Autorizējies mājaslapā ar lietotāju, kuram ir saimniecības nodaļas piekļuve 2) Skolotāji ir izveidojuši pieteikumus priekš saimniecības nodaļas 3)Skolotāji ir atzīmējuši pieteikumu “Atrisināts(Pārbaudīts)”	Ierakstot meklētājā pieteikuma informāciju	1) Iela 2) Telpa 3) Nodaļa 4) Problēma 5) Piezīme 6) Vārdu 7) Uzvārdu 8) E-pastu	Parāda atrisināto pieteikumu vēsturi	PR.19



### 6.3. Testpiemēru žurnāls

Testēšanas ID	Testpiemēra ID	Testēšanas nosaukums	Statuss	Kļūda
T-1	TP.01	Mājaslapas uzstādīšana	Veiksmīgs	
T-2	TP.02	Autorizēties sistēma ar Microsoft API	Neveiksmīgs	Nebija pareizi ievadīti Microsoft API dati
T-3	TP.02	Autorizēties sistēma ar Microsoft API	Veiksmīgs	
T-4	TP.03	Savienojums ar Microsoft	Veiksmīgs	
T-5	TP.04	Lietotājs nav autorizēts sistemā	Veiksmīgs	
T-6	TP.05	Administrācijas paneļa skats	Veiksmīgs	
T-7	TP.06	Pieteikuma atrisināšana administrācijas panelī	Veiksmīgs	
T-8	TP.07	Pieteikuma izdzēšana administrācijas panelī	Neveiksmīgs	Nebija pareizi ierakstīts SQL vaicājums
T-9	TP.07	Pieteikuma izdzēšana administrācijas panelī	Veiksmīgs	
T-10	TP.08	Pieteikuma rediģēšana administrācijas panelī	Neveiksmīgs	Nesaglabāja neizmainīto pieteikuma informāciju
T-11	TP.08	Pieteikuma rediģēšana administrācijas panelī	Veiksmīgs	
T-12	TP.09	Skolotāja sākumlapa	Veiksmīgs	
T-13	TP.10	Skolotāja sākumlapas pieteikumi	Neveiksmīgs	Rādīja visus pieteikumus, kurus nepievienoja lietotājs
T-15	TP.10	Skolotāja pieteikuma pieteikumi	Veiksmīgs	
T-16	TP.11	Skolotāja pieteikuma pievienošana	Veiksmīgs	
T-17	TP.12	Skolotāja sākumlapas pieteikuma atcelšana	Veiksmīgs	

Testēšanas ID	Testpiemēra ID	Testēšanas nosaukums	Statuss	Kļūda
<b>T-18</b>	<b>TP.13</b>	Skolotāja sākumlapas pieteikuma apstiprināšana	Veiksmīgs	
<b>T-19</b>	<b>TP.14</b>	Skolotāja sākumlapas pieteikums neizdarīts	Veiksmīgs	
<b>T-20</b>	<b>TP.15</b>	Saimniecības nodaļas dabinieka sākumlapa	Veiksmīgs	
<b>T-21</b>	<b>TP.16</b>	Saimniecības nodaļas dabinieka sākumlapas pieteikumi	Veiksmīgs	
<b>T-22</b>	<b>TP.17</b>	Saimniecības nodaļas dabinieka pieteikums izdarīts	Veiksmīgs	
<b>T-23</b>	<b>TP.18</b>	Saimniecības nodaļas dabinieka pieteikums neizdarīts	Veiksmīgs	
<b>T-24</b>	<b>TP.19</b>	Saimniecības nodaļas dabinieka pieteikums procesā	Veiksmīgs	
<b>T-25</b>	<b>TP.20</b>	Saimniecības nodaļas dabinieka atrisināto pieteikumu vēsture	Neveiksmīgs	Nebija pareizs meklētāja SQL vaicājums
<b>T-26</b>	<b>TP.20</b>	Saimniecības nodaļas dabinieka atrisināto pieteikumu vēsture	Neveiksmīgs	Kods nebija pareizi īstenots mājaslapā
<b>T-27</b>	<b>TP.20</b>	Saimniecības nodaļas dabinieka atrisināto pieteikumu vēsture	Veiksmīgs	

## **7. Individuālais ieguldījums**

Mans individuālais ieguldījums problēmpieteikumu sistēmas izstrādē Liepājas Valsts tehnikumā bija dažādas funkcijas un uzdevumi, ko es veicu.

Analīze un specifikācija: es aktīvi iesaistījos problēmu analīzē un prasību specifikāciju izstrādē. Es identificēju sistēmas funkcionalitāti, prakses vadītāja prasības, kas tika iekļautas beigu darbā..

Sistēmas modelēšana: es veicu sistēmas modelēšanas darbus, tostarp struktūras modeli, klašu diagrammas, funkcionālo un dinamisko sistēmas modeļus. Es nodrošināju, ka sistēma ir loģiski strukturēta un atbilst galvenajām prasībām.

Lietotāju ceļvedis: es izstrādāju lietotāju ceļvedi, kas nodrošina skaidru norādījumu par to, kā sistēmu lietot. Šis ceļvedis palīdzēs skolotājiem un citiem lietotājiem ērti izmantot problēmpieteikumu sistēmu.

Testēšanas dokumentācija: es sagatavoju testēšanas dokumentāciju, kas ietvēra testēšanas metodes, testpiemērus un testēšanas žurnālu. Ar šo dokumentāciju mēs varējām nodrošināt sistēmas kvalitāti un atklāt iespējamus kļūdu vai nepilnību avotus.

Iespējams dizaina un izmantojamības uzlabojumi: es devu savus ieteikumus par sistēmas izmantojamību un dizaina aspektiem, lai padarītu to vēl efektīvāku un viegli lietojamu. Manas idejas tika ņemtas vērā, lai nodrošinātu labāku lietotāja pieredzi.

## 8. Secinājumi

Darbojoties problēmpieteikumu sistēmas izstrādē Liepājas Valsts tehnikumā, esmu nonācis pie dažādiem secinājumiem. Šeit ir svarīgākie secinājumi, ko esmu guvis no šī projekta:

1. Plānošana ir nepieciešama, lai varētu izplānot savu laiku, kā arī lai zinātu visefektīvāko risinājumu projekta uzsākšanai un tā pabeigšanai.
2. Ieteikumi un uzlabojumi ir nepieciešami - skolotāju atsauksmes un ieteikumi ir ļoti vērtīgi sistēmas uzlabošanai. Esmu iemācījies novērtēt šādas atsauksmes un izstrādāt konkrētus priekšlikumus par sistēmas funkcionalitātes, izmantojamības un dizaina uzlabojumiem.
3. Veicot šo projektu, ieguvu papildus pieredzi, kā arī iemācījos efektīvi un pareizi rakstīt kodu atbalsta kodēšanu standartiem un tām prasībām.
4. Taisot problēmpieteikumu sistēmu, es iemācījos un apguvu vairākas programmēšanas valodas, kā, piemēram, CSS, HTML, PHP, JS un pieredzi ar bootstrap bibliotēku.
5. Problēmpieteikumu sistēma deva jaunu skatu uz mājaslapu programmēšanu.

Sastaptās problēmas un izaicinājumi:

1. Galvenā problēma, ar ko saskāros, bija laika plānošana un prasme darbu sadalīt divu cilvēku komandu struktūrā.
2. Bija grūtības efektīvi koordinēt uzdevumus un panākt, lai abas personas darbojas līdzīgā tempā. Rezultātā bija vajadzīgs pielāgoties, lai nodrošinātu līdzsvaru starp darbu un sasniegumiem.

Mūsu nākamais solis ir veikt vairāk testēšanu un novērtēt sistēmas darbību reālajā vidē. Vēlos uzlabot sistēmas efektivitāti, veikt drošības pārbaudes un ņemt vērā lietotāju atsauksmes, lai attīstītu un uzlabotu Problēmpieteikumu sistēmu. Kopumā mans ieguldījums sistēmas izstrādē deva labus rezultātus, bet arī radīja dažas izaicinājumu situācijas.

## 9. Lietoto saīsinājumu saraksts

2. tabula

Terminu skaidrojums

Termins un saīsinājums	Skaidrojums
DB	Datubāze jeb DB ir informācijas kopums ar noteiktu struktūru. Par datubāzi var saukt jebkuru strukturētas informācijas kopumu.
Serveris	Serveris ir datorsistēma klienta servera arhitektūrā, kas nodrošina klientu pieprasījumu apstrādi un nosūta tiem atbildes. Terminu "serveris" attiecina gan uz datoru, gan uz programmatūru, kas veic šādas funkcijas.
Mājaslapa	"Mājaslapa" latviešu valodā ir interneta vietne vai tīmekļa lapa, kas pieejama internetā un satur informāciju, resursus, pakalpojumus vai citu saturu, ko var aplūkot, lasīt vai izmantot interneta lietotāji.
Frontend	Frontend, jeb arī priekšgals, ir tā daļa no informācijas tehnoloģijām, kas attiecas uz lietotāja saskarnes izstrādi un darbību vietnē vai aplikācijā. Priekšgals ir atbildīgs par lietotāja pieredzes (User Experience, UX) veidošanu un ieviešanu, tostarp dizainu, interaktīvo elementu izstrādi, lietotāja ievades apstrādi un vizuālo prezentāciju.
BackEnd	Backend, jeb arī aizgals, ir tā daļa no informācijas tehnoloģijām, kas atrodas aiz priekšgala, un nodrošina servera pusi, kas atbild par datu apstrādi, datu bāzu savienošanu, biznesa loģikas implementāciju un citas sistēmas darbības, kas notiek aiz aizkulisēm. Aizgals nodrošina informācijas apstrādi un datu pārvaldību, kas ir nepieciešama, lai aplikācija vai vietne darbotos pareizi un efektīvi.
CRUD	CRUD ir akronīms, kas nozīmē Izveidot (Create), Lasīt (Read), Labot (Update) un Dzēst (Delete). Tas attiecas uz pamata darbībām, kas tiek veiktas ar datiem datu bāzē <ul style="list-style-type: none"><li>• Izveidot: Tas ietver jaunu datu vai ierakstu pievienošanu sistēmai.</li><li>• Lasīt: Tas ļauj iegūt vai piekļūt esošajiem datiem sistēmā.</li><li>• Labot: Tas ļauj modificēt vai rediģēt esošos datus sistēmā.</li><li>• Dzēst: Tas ietver nevēlamu datu noņemšanu vai dzēšanu no sistēmas.</li></ul>

## 10. Literatūras un informācijas avotu saraksts

1. Build API clients for php with microsoft identity authentication  
<https://learn.microsoft.com/en-us/openapi/kiota/tutorials/php-azure?tabs=portal>
2. PHP:PDO - Manual  
<https://www.php.net/manual/en/book.pdo.php>
3. Bootstrap document  
<https://getbootstrap.com/docs/5.3/getting-started/introduction/>
4. Saite uz sistēmas pirmkoda  
<https://github.com/KnsGoyoLV/SchoolTicket>

```
// searchbar query stuff that search for stuff
if (isset($_POST['searchbar']))
    $keyword = $_POST['searchbar'];

if (isset($keyword) && !empty($keyword)) {
    $result = $pdo->query("SELECT * FROM pieteikums where telpa like '%$keyword%' or status like '%$keyword%' or iela like '%$keyword%' or problema
        like '%$keyword%' or piezimes like '%$keyword%' or epasts like '%$keyword%' ORDER BY `pieteikums`.`laiks` DESC");
} else {
    $result = $pdo->query("SELECT * FROM pieteikums ORDER BY `pieteikums`.`laiks` DESC");
}
// get count of every status tickets
$done = $pdo->query("SELECT * FROM pieteikums where (status = 'Atrisināts') or (status = 'Atrisināts(Parbaudīts)') ");
$notdone = $pdo->query("SELECT * FROM pieteikums where (status = 'Neatrisināts')");
$proces = $pdo->query("SELECT * FROM pieteikums where (status = 'Procesā')");

if (isset($_GET['total']) && $_GET['total'] === 'true') {
    $rows = $result->fetchAll();
}
if (isset($_GET['done']) && $_GET['done'] === 'true') {
    $rows = $done->fetchAll();
}
if (isset($_GET['notdone']) && $_GET['notdone'] === 'true') {
    $rows = $notdone->fetchAll();
}
if (isset($_GET['process']) && $_GET['process'] === 'true') {
    $rows = $proces->fetchAll();
}

?>
```

### 13.attēls. koda attēls

```
function Invalid_session($email)
{
    if (!isset($_SESSION['t'])) {
        header('location:login.php');
    }
    $admin = array(
        'daniels.vidopskis@sk.lvt.lv1'
    );
    if (in_array($_SESSION['email'], $admin)) {
        header("location:../tasker/task.php");
    }
}
```

### 14.attēls. koda attēls

```

function MicrosoftInfo()
{

    $env = parse_ini_file('../database/.env');
    session_start();

    $tenant = $env['tenant'];
    $client_id = $env['client_id'];
    $client_secret = $env['client_secret'];
    $callback = $env['callback'];

    if (array_key_exists('access_token', $_POST)) {
        //save access_token to SESSION t variable
        $_SESSION['t'] = $_POST['access_token'];
        $t = $_SESSION['t'];
        $ch = curl_init();
        //get users json data from /me/ endpoint
        curl_setopt($ch, CURLOPT_HTTPHEADER, array(
            'Authorization: Bearer ' . $t,
            'Content-type: application/json'
        ));
        curl_setopt($ch, CURLOPT_URL, "https://graph.microsoft.com/v1.0/me/");
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $rez = json_decode(curl_exec($ch), 1);
        if (array_key_exists('error', $rez)) {
            //if we have a error dump everything that error gives us
            var_dump($rez['error']);
            die();
        } else { // get users data from json with $rez variable name
            $_SESSION['username'] = $rez["givenName"];
            $_SESSION['surname'] = $rez["surname"];
            $_SESSION['email'] = $rez['mail'];
            $_SESSION['id'] = $rez["id"];
            $_SESSION['job'] = $rez["jobTitle"];
            $_SESSION['fullname'] = $rez["displayName"];
        }
        curl_close($ch);
        header('Location: ' . $callback);
    }
    session_abort();
}

```

## 15.attēls. koda attēls

```
// Initialize the required fields array
$required = array('nodala', 'Problēma', 'Telpa', 'Iela');

// Initialize an empty error array
$errors = array();

// Check if the form has been submitted
if (isset($_POST['submit1'])) {
    // Loop over the required fields and check for empty values
    foreach ($required as $field) {
        if (empty($_POST[$field])) {
            $errors[] = $field . ' lauks ir obligāts';
        } else {
            // Store the submitted value in the data array
            $data[$field] = $_POST[$field];
        }
    }
}
```

**9.attēls. koda attēls**

```
// Display the error messages (if any)
if (!empty($errors)) {
    echo '<div class="alert alert-danger" role="alert">';
    echo 'Lūdzu aizpildiet sekojošus laukus: ';
    echo implode(', ', $errors);
    echo '</div>';
}
function is_selected($value, $selected)
{
    return $value === $selected ? 'selected' : '';
}
```

**10.attēls. koda attēls**