Liepājas Valsts tehnikums

Tīmekļu vietne “AnimalHouse”

Kvalifikācijas eksāmena praktiskās daļas dokumentācija

Profesionālā kvalifikācija Programmēšanas tehniķis

Grupas nosaukums 4PT

Projekta izstrādātājs Daniils Ternoviks

/vārds, uzvārds, paraksts/

Eksāmena datums 2024. gada 20.junījs

Liepāja 2024

**Satura rādītājs**

[Ievads 14](#_Toc168653689)

[1. Uzdevuma formulējums 15](#_Toc168653690)

[2. Programmatūras prasību specifikācija 16](#_Toc168653691)

[2.1. Produkta perspektīva 16](#_Toc168653692)

[2.2. Sistēmas funkcionālās prasības 16](#_Toc168653693)

[2.3. Sistēmas nefunkcionālās prasības 28](#_Toc168653694)

[3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums 29](#_Toc168653695)

[3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts 30](#_Toc168653696)

[3.2. Iespējamo (alternatīvo) risinājuma līdzekļu un valodu apraksts 31](#_Toc168653697)

[4. Sistēmas modelēšana un projektēšana 32](#_Toc168653698)

[4.1. Sistēmas struktūras modelis 32](#_Toc168653699)

[4.2. ER diagramma 33](#_Toc168653702)

[4.3. Funkcionālais un dinamiskais sistēmas modelis 35](#_Toc168653703)

[4.3.1. Datu plusmu diagramma dzīvniekam. 35](#_Toc168653705)

[4.3.3. Datu plusmu diagramma administratoram. 36](#_Toc168653710)

[4.4. Aktivitāšu diagramma (Activity) 37](#_Toc168653713)

[4.4.1. Aktivitāšu diagramma - 1.daļa 37](#_Toc168653715)

[4.4.2. Aktivitāšu diagramma - 2.daļa 38](#_Toc168653718)

[4.5. Lietojumgadījumu diagramma (Use Case) 39](#_Toc168653720)

[4.6. Sistēmas moduļu apraksts un algoritmu shēmas 40](#_Toc168653723)

[5. Lietotāju ceļvedis 41](#_Toc168653724)

[5.1. Mājas lapas reģistrācijas sadaļas izmantošana 41](#_Toc168653726)

[5.2. Mājas lapas autorizācijas sadaļas izmantošana 43](#_Toc168653727)

[5.3. Mājas lapas dzīvnieku tabulas izmantošana un apskatīšana 45](#_Toc168653730)

[5.4. Dzīvnieku pievienošana 46](#_Toc168653733)

[5.5. Esoša dzīvnieka datu rediģēšana 47](#_Toc168653738)

[5.6. Dzīvnieku bildes pievienošana 49](#_Toc168653740)

[5.7. Lietotāju tabulas apskatīšana 50](#_Toc168653744)

[5.8. Lietotāja datu rediģēšana 51](#_Toc168653745)

[5.9. Jauna lietotāja izveidošana 52](#_Toc168653746)

[5.10. Profila atveršana 53](#_Toc168653747)

[6. Testēšanas dokumentācija 55](#_Toc168653748)

[6.1. Izvēlētās testēšanas metodes, rīku apraksts un pamatojums 55](#_Toc168653749)

[6.2. Testpiemēru kopa 58](#_Toc168653750)

[6.3. Testēšanas žurnāls 64](#_Toc168653755)

[8. Secinājumi 65](#_Toc168653756)

[9. Lietoto terminu un saīsinājumu skaidrojumi 67](#_Toc168653757)

[10. Literatūras un informācijas avotu saraksts 68](#_Toc168653758)

[PIELIKUMS 69](#_Toc168653759)

# Ievads

Šīs projekts, “Dzīvnieku māja”, ir mans mēģinājums padarīt dzīvnieka iegūšanu par digitālo risinājumu, ar mērķi veicināt dzīvnieka pieejamību, kā arī veidot lielu dzīvnieku pārdošanas platformu, “Dzīvnieku māja” ir liela un ērta platforma, kurā lietotāji var piedāvāt savus mājdzīvniekus, kā arī atrast sev vajadzīgo un interesantu sev dzīvnieku un iegādāties viņu.

Šajā dokumentā iekļauta informācija par:

* Programmatūras prasību specifikāciju, kas nosaka funkcionālas prasības.
* Izstrādes līdzekļu, rīku aprakstu un izvēles pamatojumu, kās skaidro tehniskos lēmumus un risinājumu izvēles loģiku.
* Sistēmas modelēšanu un projektēšanu, sniedzot pārskatu par dzīvnieku pārdošanas platformu.
* Lietotāju ceļvedi, kas palīdzēs izprast, kā lietotāji izmantos un mijiedarbosies ar platformu.
* Testēšanas dokumentāciju, kas nodrošina platformas uzticamību un drošību.
* Secinājumus, kas atspoguļo darba rezultātus un projektu kopuma.

Projekta pamatuzdevumi ietver dzīvnieku reģistrēšanas un pārvaldības procesu automatizāciju, nodrošinot lietotājiem ērtu rīku klāstu dzīvnieku augšupielādei, dzīvnieku aprakstiem, kā arī dzīvnieka cenu noteikšanai. Platformā tiek uzsvērta drošība un datu precizitāte, lai lietotāji ir pārliecināti par katru no dzīvniekiem un to vērtību.

# 1. Uzdevuma formulējums

Projekta "Dzīvnieku māja" galvenais mērķis ir izstrādāt un integrēt digitālu risinājumu, kas pilnveido un apvieno visas būtiskās funkcionalitātes, saistītas ar dzīvnieku pirkšanu, pārdošanu un apkopi. Šis projekts izceļas ar viegli saprotamu un efektīvu funkcionalitāti, nodrošinot lietotājiem ērtu izmantošanas pieredzi un daudzefektīvus rīkus.

Lietotājiem ir iespēja augšupielādēt informāciju par savu dzīvnieku uz mūsu platformas tabulas, sniedzot visu nepieciešamo informāciju. Katram lietotājam ir personīgais profils, kurā apkopota informācija par viņu, un piedāvājums apskatīt savus pirkumus vai sludinājumus.

Platforma piedāvā lietotājiem plašu dzīvnieku izvēli, izmantojot detalizētu tabulu ar iespēju izvēlēties dzīvniekus, piemēram, kaķus, suņus vai citus. Katram dzīvniekam ir iespēja pievienot attēlus, un platforma nodrošina drošu pirkšanas procesu.

Visas šīs funkcijas tiek izstrādātas tā, lai gan pieredzējušie lietotāji, gan jaunpienācēji varētu viegli izmantot platformu, izmantojot daudzefektīvus rīkus pirkumiem vai pārdošanai.

# 2. Programmatūras prasību specifikācija

Šajā nodaļā jūs redzēsiet informāciju par produkta perspektīvu, sistēmas funkcionālām prasībām, kā arī nefunkcionālām prasībām.

## 2.1. Produkta perspektīva

"Dzīvnieku māja" - tas ir unikāls tiešsaistes dzīvnieku audzētava, kas sniedz izcilas iespējas tiem, kuri meklē ne tikai ideālu mājdzīvnieku, bet arī dziļu mijiedarbību ar dzīvnieku pasauli. Mūsu projekts balstās uz rūpīgu partneru atlasi un vēlmi piedāvāt mūsu lietotājiem neaizmirstamu pieredzi, izvēloties un rūpējoties par saviem nākamajiem draugiem.

## 2.2. Sistēmas funkcionālās prasības

**P.1. Reģistrēšana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju lietotājam piereģistrēties “Dzīvnieku māja” mājaslapā

Ievaddati:

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligāts | Piezīmes |
| username | jā | jābūt unikālam |
| password | jā | Vismāz 8 simboli |

Apstrāde:

1) Mājaslapa pārbauda vai dati ir ievadīti korekti

2) Mājaslapa pārbauda vai ievadītais vārds nav aizņemts

3) Mājaslapa pārbauda vai ievadīta parole ir vismāz 8 simboli

Izvaddati:

Lietotājs tiek pārvietots uz Autorizēšanas logu.

**P.2. Autorizēšana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju lietotājam autorizēties “Dzīvnieku māja” mājaslapā

Ievaddati:

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligāts | Piezīmes |
| username | jā | nav |
| password | jā | nav |

Apstrāde:

1) Mājaslapa pārbauda vai dati ir ievadīti korekti

2) Mājaslapa pārbauda vai ievadītais vārds eksistē

3) Mājaslapa pārbauda vai parole ir ievadīta korekti un sakrīt ar lietotājvārdu reģistrēto

Izvaddati:

Lietotājs tiek pārvietots uz sākuma lapu.

**P.3. Dzīvnieku pievienošana “Dzīvnieku māja” mājaslapas tabulā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram pievienot mājaslapas tabulā jauno dzīvnieku

Ievaddati:

1. Atvērta mājaslapa
2. Atvērta tabula “Table with animals”
3. Lietotāja statuss ir Administrators vai moderators
4. Blakus tabulai nospiesta poga “Create”

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligāts | Piezīmes |
| Name | jā | nav |
| Type | jā | nav |
| Date of birth | jā | Ierakstīt ar cipariem vai izvelēties izmantojot date picker |
| Gender | jā | Izvelēties ar dropdown |

Apstrāde:

1) Mājaslapa pārbauda vai dati ir ievadīti korekti

2) Mājaslapa pārbauda dzīvnieka ID kuru vajadzēs rediģēt

Izvaddati:

Tiek pievienots jauns dzīvnieks.

**P.4. Dzīvnieku dzešana no “Dzīvnieku māja” mājaslapas tabulā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram dzēst mājaslapas tabulā esošo dzīvnieku

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Table with animals”

3) Lietotāja statuss ir Administrators vai Moderators

4) Tabulas laukā nospiestā poga “Delete”

Apstrāde:

1. Mājaslapa pārbauda dzīvnieka ID kuru vajadzēs dzēst
2. Mājaslapa pārbauda vai lietotaja statuss ir Administrators vai moderators

Izvaddati:

Tiek dzēsts esošais dzīvnieks.

**P.5. Dzīvnieka datu izlabošana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram izlabot datus mājaslapas tabulā izvelētajam dzīvniekam.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Table with animals”

3) Lietotāja statuss ir Administrators vai Moderators

4) Tabulas laukā nospiestā poga “Edit”

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligāts | Piezīmes |
| Name | jā | nav |
| Type | jā | nav |
| Date of birth | jā | Ierakstīt ar cipariem vai izvelēties izmantojot date picker |
| Gender | jā | Izvelēties ar dropdown |

Apstrāde:

1) Mājaslapa pārbauda dzīvnieka ID kuru vajadzēs dzēst

2) Mājaslapa pārbauda vai dati ir ievadīti korekti

Izvaddati:

Dzīvnieka dati tiek rediģēti.

**P.6. Dzīvnieka bildes pievienošana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram pievienot bildes izvelētajam dzīvniekam

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Table with animals”

3) Lietotāja statuss ir Administrators vai Moderators

4) Tabulas laukā nospiestā poga ar bildes zīmi

Apstrāde:

1) Mājaslapa pārbauda dzīvnieka ID kurām vajadzēs pievienot bildi

Izvaddati:

Dzīvniekam tiek pievienotas bildes.

**P.7. Lietotāja datu izlabošana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram izlabot nepieciešamos datus lietotājam

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Table with users”

3) Lietotāja statuss ir Administrators vai Moderators

4) Tabulas laukā nospiestā poga “Edit”

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligats | Piezimes |
| username | jā | nav |
| password | jā | nav |
| role | jā | Izvelēties ar dropdown |

Apstrāde:

1) Mājaslapa pārbauda vai dati ir ievadīti korekti

2) Mājaslapa pārbauda lietotāja ID kurām būs jāmaina datus

Izvaddati:

Lietotājam tiek izmainīti dati.

**P.8. Lietotāja statusa rediģēšana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram izmainīt tabulā lietotāja statusu

Ievaddati:

1) Atvērta mājaslapa

2) Lietotāja statuss ir Administrators vai Moderators

3) Tabulā “Table with users” nospiestā poga “Edit”

4) Ievadīti nepieciešamie dati

5) Nospiesta poga “Save”

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligats | Piezimes |
| username | jā | nav |
| password | nē | nav |
| role | jā | Izvelēties ar dropdown |

Apstrāde:

1) Mājaslapa pārbauda lietotāja ID kurām būs jāmaina statusu

Izvaddati:

Tiek izmainīts lietotāja statuss.

**P.9. Lietotāja izveidošana “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju Administratoram vai Moderatoram izveidot tabulā jauno lietotāju

Ievaddati:

1) Atvērta mājaslapa

2) Lietotāja statuss ir Administrators vai Moderators

3) Tabulā “Table with users” nospiestā poga “Create”

4) Ievadīti nepieciešamie dati

5) Nospiesta poga “Save”

|  |  |  |
| --- | --- | --- |
| Nosaukums | Obligats | Piezimes |
| username | jā | nav |
| password | nē | nav |
| role | jā | Izvelēties ar dropdown |

Apstrāde:

1) Mājaslapa pārbauda vai dati tiek ierakstīti korekti

Izvaddati:

Tiek pievienots jauns lietotājs.

**P.10. Pieprasījuma nosūtīšana uz dzīvnieka publicēšanu “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju visiem lietotājiem aizsūtīt pieprasījumu uz dzīvnieku publicēšanu dzīvnieku kataloga

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Animals”

3) Nospiesta poga “Сreate a query”

4) Ievadīti nepieciešamie dati

5) Nospiesta poga “Send”

Apstrāde:

1) Mājaslapa pārbauda vai dati tiek ierakstīti korekti

2) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

Izvaddati:

Pieprasījums tiek nosūtīts.

**P.11. Rādīt dzīvnieka katalogu “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju visiem lietotājiem redzēt katalogu ar visiem atļautiem dzīvniekiem.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta lapa “Animals”

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

Izvaddati:

Katalogs tiek parādīta.

**P.12. Dzīvnieku filtrēšanas kataloga “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju visiem lietotājiem filtrēt dzīvniekus pec nepieciešamiem datiem.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta lapa “Animals”

3) Pirmajā kartē nospiesta poga ar vajadzīgiem datiem

Apstrāde:

2) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

3) Mājaslapa pārbauda kāds parametrs ir izvelēts

Izvaddati:

Katalogs tiek filtrēts.

**P.13. Lietotāja personīgais profils “Dzīvnieku māja” mājaslapā**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam ieiet savā personīgajā profilā un apskatīties informāciju par saviem pirkumiem, pāsutījumu vēsturi un apskatīt savus datus.

Ievaddati:

1) Atvērta mājaslapa

2) Nospiesta poga “Profile”

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

Izvaddati:

Lietotāja profils tiek parādīts.

**P.14. Iespēja nosūtiet e-pasta vēstuli vietnes administrācijai**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam aizsutīt e-pasta vēstuli administrācijai.

Ievaddati:

1) Atvērta mājaslapa

2) Tiek aiziets uz mājaslapas beigām

3) Nospiesta poga “Send mail”

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

2) Mājaslapa pārbauda vai lietotājs ierakstījis visu informaciju.

3) Mājaslapa pārbauda vai tiek nospiesta poga “Send”

Izvaddati:

Vēstule tiek nosūtīta.

**P.15. Iespēja atrāst mūs piedāvātos socialajos tīklos**

Mērķis:

Funkcija nodrošina iespēju katrām lietotājam apskatīties mūs arī sociālajos tīklos.

Ievaddati:

1) Atvērta mājaslapa

2) Aiziets uz mājaslapas apakšnodaļu

3) Nospiesta poga ar vajadzīgo sociālo tīklu

Apstrāde:

1) Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

Izvaddati:

Lietotājs tiek aizsūtīts uz izvēlēto sociālo tīklu.

**P.16. Iespēja filtrēt dzīvniekus pec nepieciešamiem datiem “Dzīvnieku māja” mājaslapās dzīvnieku tabulā**

Mērķis:

Funkcija nodrošina iespēju administratoram vai moderatoram filtrēt dzīvniekus tabulā pec nepieciešamiem datiem.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Table with animals”

3) Nospiests trīsstūris blakus vajadzīgiem parametriem

Apstrāde:

1. Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā
2. Mājaslapa pārbauda vai lietotājam ir statuss Administrators vai Moderators
3. Mājaslapa pārbauda kāds parametrs tiek izvelēts

Izvaddati:

Tabula tiek filtrēta.

**P.17. Iespēja filtrēt lietotājus pec nepieciešamiem datiem “Dzīvnieku māja” mājaslapās lietotāju tabulā**

Mērķis:

Funkcija nodrošina iespēju administratoram vai moderatoram filtrēt lietotājus tabulā pec nepieciešamiem datiem.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērta tabula “Table with users”

3) Nospiests trīsstūris blakus vajadzīgiem parametriem

Apstrāde:

1. Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā
2. Mājaslapa pārbauda vai lietotājam ir statuss Administrators vai Moderators
3. Mājaslapa pārbauda kāds parametrs tiek izvelēts

Izvaddati:

Tabula tiek filtrēta.

**P.18. Iespēja izlogoties no sistēmas**

Mērķis:

Funkcija nodrošina iespēju jebkurām lietotājam izlogoties no sistēmas.

Ievaddati:

1) Atvērta mājaslapa

2) Lietotājs ir ielogots sistēmā

3) Nospiesta poga “Logout”

Apstrāde:

1. Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā

**P.19. Iespēja apskatīties detalizēto informāciju par dzīvnieku**

Mērķis:

Funkcija nodrošina iespēju lietotājam apskatīties detalizēto informaciju par dzīvnieku katalogā.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērts katalogs “Animals”

3) Pie izvelēta dzīvnieka nospiesta poga “Read more”

Apstrāde:

1. Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā
2. Mājaslapa pārbauda kāds dzīvnieks ir izvelēts

Izvaddati:

Detalizēta informācija par dzīvnieku tiek izvadīta.

**P.20. Iespēja apmaksāt dzīvnieku**

Mērķis:

Funkcija nodrošina iespēju lietotājam apmaksāt izvelēto dzīvnieku.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērts katalogs “Animals”

3) Pie izvelēta dzīvnieka nospiesta poga “Read more”

4) Atvertā logā nospiest pogu “Buy”

5) Ievadīt visu vajadzīgo informāciju

Apstrāde:

1. Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā
2. Mājaslapa pārbauda kāds dzīvnieks ir izvelēts
3. Mājaslapa pārbauda vai visi dati ir korekti ievadīti

Izvaddati:

Dzīvnieks tika nopirkts.

**P.21. Iespēja lietotājam izmainīt savu paroli**

Mērķis:

Funkcija nodrošina iespēju lietotājam izmainīt savu paroli.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērts profils “Profile”

3) Nospiesta poga “Change password”

4) Ievadīt jauno paroli

5) Ievadīt jauno paroli velreiz lai apstiprināt darbības

Apstrāde:

1. Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā
2. Vai paroles sakrīt
3. Vai parolē ir vismaz 8 simboli

Izvaddati:

Parole tiek izmainīta.

**P.22. Iespēja lietotājam izmainīt savu lietotājvardu**

Mērķis:

Funkcija nodrošina iespēju lietotājam izmainīt savu lietotājvardu.

Ievaddati:

1) Atvērta mājaslapa

2) Atvērts profils “Profile”

3) Nospiesta poga “Change username”

4) Ievadīt jauno lietotājvardu

Apstrāde:

1. Mājaslapa pārbauda vai lietotājs ir ielogots sistēmā
2. Mājaslapa pārbauda vai lietotājvards nav aizņemts

Izvaddati:

Lietotājvards tika izmainīts.

## 2.3. Sistēmas nefunkcionālās prasības

1) Programmatūras izstrādes procesa dokumentācijai ir jābūt noformētai un izstrādātai atbilstoši Latvijas Valsts standartam LVS 68:1996 par programmatūras prasību specificēšanu.

2) Lietotāju saskarnei ir jābūt Angļu valodā.

3) Mājaslapas dominējošas krāsām ir jābūt sekojošiem: Zils, Balts, Pelēks.

4) Mājaslapas saskarnei ir jābūt ērtai un ergonomiskai (tādai, kas minimizē IS lietotāja slodzi, piemēram, viegli uztveramai).

# 3. Izstrādes līdzekļu, rīku apraksts un izvēles pamatojums

Izstrādes līdzekļi un rīki, kas tika izmantoti šim projektam, ir rūpīgi atlasīti, lai nodrošinātu efektīvu un integrētu digitālo risinājumu.

1. **Backend izstrāde (Java):**

* Java tika izvēlēta kā backend valoda, jo tā piedāvā plašas iespējas un ir pazīstama ar savu uzticamību un drošību.
* Java nodrošina lielisku veiktspēju un skalējamību, kas ir būtiski digitālajiem risinājumiem ar pieaugošu apmeklētību.

1. **Frontend izstrāde (React un TSX):**

* React un TypeScript (TSX) tika izvēlēti frontend izstrādei, jo tie nodrošina modulāru un viegli uztveramu kodu.
* React piedāvā efektīvas komponentes un atjaunošanu tikai nepieciešamajām daļām, kas uzlabo lietotāja pieredzi un veiktspēju.

1. **Datu bāze (PostgreSQL un pgAdmin4):**

* PostgreSQL tika izvēlēts kā datu bāzes pārvaldības sistēma, jo tas ir atvērtā pirmkoda un piedāvā plašas iespējas attiecībā uz datu modeļiem un skaļojamību.
* pgAdmin4 ir izmantojams kā vizuāls rīks datu bāzes pārvaldībai, nodrošinot ērtu un pārredzamu interfeisu.

1. **API saskarne (Postman):**

* Postman tiek izmantots kā rīks, lai veiktu un pārbaudītu HTTP pieprasījumus uz serveri.
* Tas piedāvā intuitīvu lietošanas pieredzi un palīdz uzlabot un pārbaudīt backend API funkcionalitāti.

**Izvēles pamatojums:**

* Java un React ir populāras un plaši izmantotas tehnoloģijas ar aktīvu attīstības kopienas atbalstu.
* PostgreSQL piedāvā attiecības datu bāzes priekšrocības, kā arī atbilst digitālo risinājumu augstajiem prasību standartiem.
* Postman sniedz ērtu veidu, kā pārbaudīt un optimizēt servera API izsaukumus, nodrošinot to efektivitāti un drošību.

Šie izvēlētie līdzekļi un rīki kopā veido integrētu un efektīvu risinājumu, kas nodrošina lietotājiem ērtu un drošu pieredzi, vienlaikus nodrošinot uzticamu sistēmas darbību un datu integritāti.

## 3.1. Izvēlēto risinājuma līdzekļu un valodu apraksts

**Backend izstrāde (Java):**

Java ir vispār atpazīta un augsti izvērtēta programmatūras valoda, kas tika izvēlēta kā backend valoda šim projektam. Tā piedāvā stabilu un drošu vidi, un tā ir ļoti piemērota lielāku projektu izstrādei. Ar bagātīgu standart bibliotēku un atbalstu plašai kopienai, Java nodrošina iespēju veiksmīgi veidot kompleksus backend risinājumus. Šī valoda ir ievērojama arī ar to, ka tā ļauj izveidot drošus un efektīvus serverus, kas ir būtiski digitālās platformas darbībai.

**Frontend izstrāde (React un TSX):**

React, kombinēts ar TypeScript (TSX), nodrošina modernu un modulāru pieeju lietotāja saskarnei. React piedāvā komponentu struktūru, kas uzlabo kodu uztveramību un atjauno tikai nepieciešamās daļas, palielinot lietotāja pieredzi. TypeScript sniedz papildu drošību, lai novērstu dažādas kļūdas jau izstrādes posmā. Šis izvēlētais kombinācijas risinājums ir īpaši piemērots projektam, kurā lietotāja saskarne ir būtiska un nepieciešama augsta veiktspēja.

**Datu bāze (PostgreSQL un pgAdmin4):**

PostgreSQL tiek izmantots kā datu bāzes pārvaldības sistēma, jo tā piedāvā attiecību datu bāzes priekšrocības un elastību datu modeļu definēšanā. Ar aktīvu attīstību un plašu kopienas atbalstu PostgreSQL nodrošina uzticamu datu uzglabāšanas sistēmu. pgAdmin4 ir vizuāls rīks, kas atvieglo datu bāzes pārvaldību, piedāvājot pārredzamu sakārni un uzlabojot administratīvās funkcijas.

**API saskarne (Postman):**

Postman ir plaši izmantots rīks, ko izmanto izstrādātāji, lai testētu un pārbaudītu API saziņu. Tas nodrošina iespēju izsaukt un pārbaudīt dažādus servera API punktus, sniedzot iespēju ērti novērtēt to darbību un efektivitāti. Šis rīks palīdz nodrošināt API drošību un pareizu funkcionēšanu.

Kopumā izvēlētie risinājuma līdzekļi ir uzmanīgi atlasīti, ņemot vērā to, kā katrs no tiem nodrošina drošību, veiktspēju un ērtību izstrādes procesā, veidojot integrētu un efektīvu digitālo platformu "Dzīvnieku māja".

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

**Alternatīvās risinājuma līdzekļi un valodas:**

**Backend izstrāde:**

Node.js un Express:

* Node.js nodrošina ātru un efektīvu servera izstrādi, balstoties uz JavaScript, un tas ir īpaši piemērots lietotāja draudzīgu un mazāk sarežģītu backend sistēmu izveidei.
* Express ir minimāls un elastīgs servera rīks, kas papildina Node.js, piedāvājot ātru izstrādi un modulāru kodu.

**Frontend izstrāde:**

Vue.js un JavaScript:

* Vue.js ir moderna un viegli uztverama JavaScript bibliotēka, kas nodrošina lietotājiem elastīgu un viegli pielāgojamu sakārni.
* Tā kā Vue.js ir viegli integrējams un saprotams, tas var būt laba alternatīva, īpaši, ja nepieciešama ātra saskarnes izstrāde.

**Datu bāze:**

MongoDB un Robo 3T:

* MongoDB piedāvā dokumentu orientētu datu bāzi, kas ir īpaši piemērota datu struktūrām, kurām bieži mainās forma.
* Robo 3T ir vizuāls rīks MongoDB pārvaldībai, piedāvājot intuitīvu saskarni un iespēju viegli rediģēt un pārvaldīt dokumentus.

**API saskarne:**

Insomnia:

* Insomnia ir alternatīvs rīks API pārbaudei un testēšanai, sniedzot intuitīvu saskarni un efektīvu veidu, kā veikt un pārbaudīt HTTP pieprasījumus.

**Iespējamais (alternatīvais) pamatojums:**

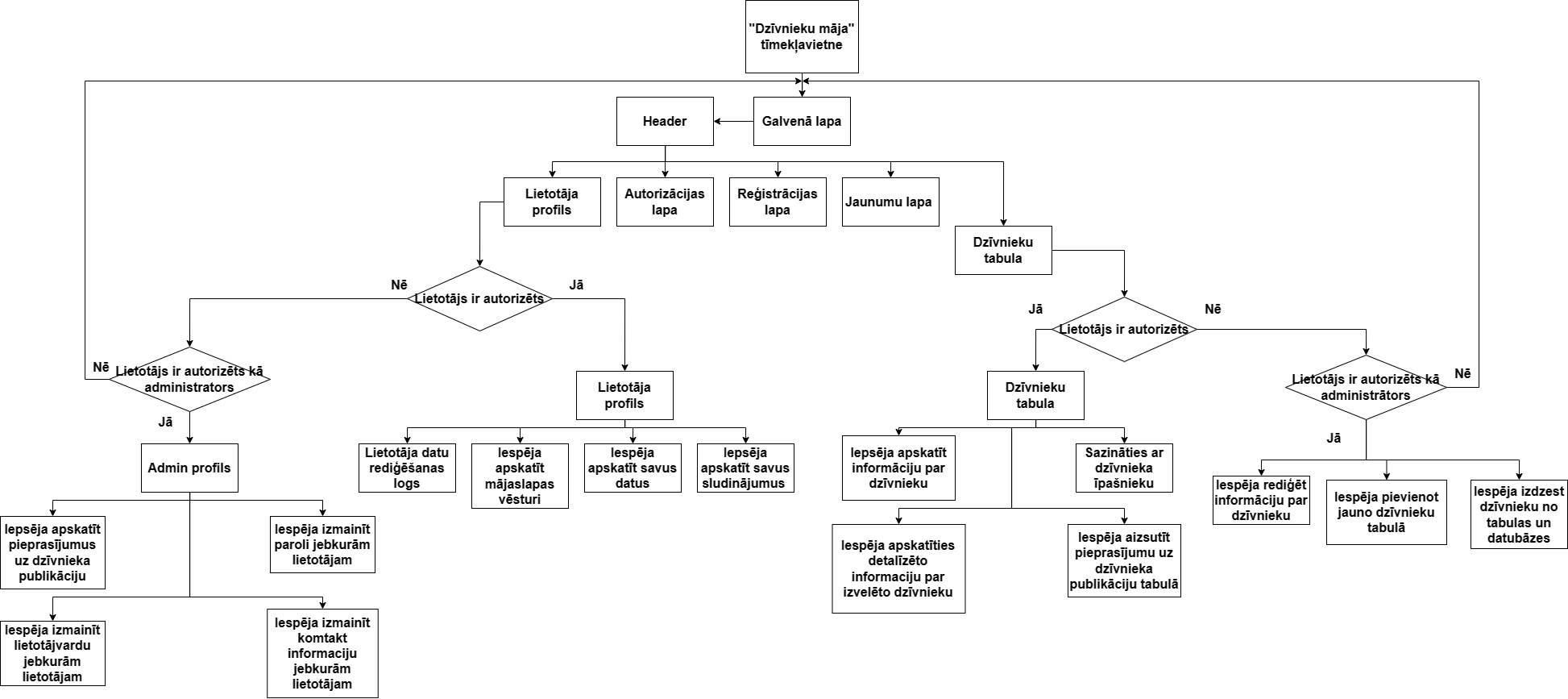
* Alternatīvie risinājumi ir izvēlēti, ņemot vērā vieglumu un ātrumu izstrādes procesā, īpaši, ja nepieciešama ātra prototipēšana vai mazākas sistēmas izstrāde.
* Vue.js un JavaScript var būt piemērota alternatīva, ja nepieciešama viegli pielāgojama un saprotama lietotāja saskarne, kas var būt īpaši noderīga nelielām projektu sākum versijām.
* MongoDB un Robo 3T nodrošina alternatīvu pieeju datu bāzes izvēlei, īpaši, ja nepieciešams elastīgs datu modelis.
* Insomnia ir viegli izmantojams rīks API saskarņu pārbaudei un testēšanai, sniedzot vienkāršu un efektīvu veidu, kā nodrošināt API funkcionalitāti.

# 4. Sistēmas modelēšana un projektēšana

Šajā nodaļā tiks apskatīta mūsu vietnes sistēmas modelēšana un projektēšana, izmantojot dažādas diagrammas, lai labāk izprastu tās darbību un funkcionalitāti. Kādas diagrammas tiek izmantotas:

1. Sistēmas struktūras modelis.
2. ER diagramma.
3. Funkcionālais un dinamiskais sistēmas modelis.
4. Aktivitāšu diagramma (Activity).
5. Lietojumgadījumu diagramma (Use Case).
6. Sistēmas moduļu apraksts un algoritmu shēmas.

## 4.1. Sistēmas struktūras modelis

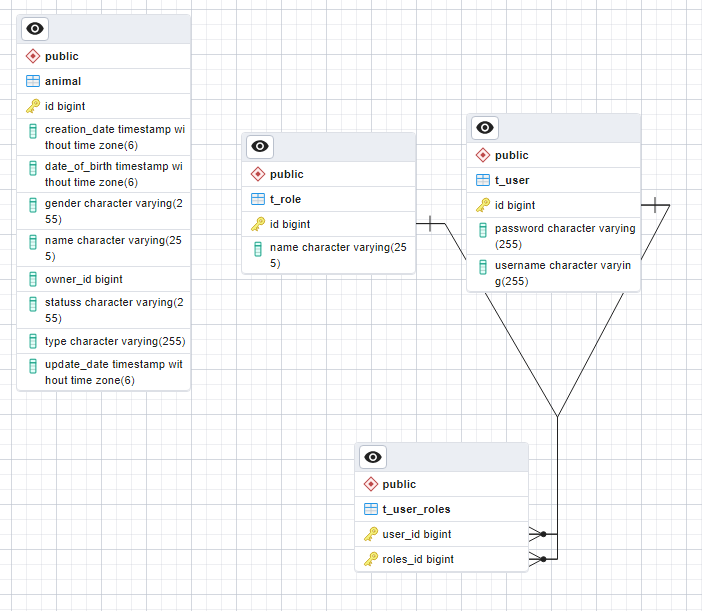
Šajā sadaļā tiks izstrādāts un aprakstīts sistēmas struktūras modelis, kas ietver visus galvenos komponentus un to attiecības sistēmā.

## 

## 1.attēls. Sistēmas struktūras modelis

## 4.2. ER diagramma

Šajā sadaļā tiks izstrādāta entīšu-sakaru (ER) diagramma, kas nodrošina informāciju par datu struktūru sistēmā un attiecībām starp datu elementiem.



**2.attēls. ER diagramma**

## 4.3. Funkcionālais un dinamiskais sistēmas modelis

## Šeit tiks apskatīts gan sistēmas funkciju apraksts, gan arī tās dinamiskais uzvedības modelis, iekļaujot procesu izpildes secību un mijiedarbību starp komponentiem.

## 4.3.1. Datu plusmu diagramma dzīvniekam.

## 

## 3.attēls. Datu plusmu diagramma dzīvniekam.

## 4.3.2. Datu plusmu diagramma lietotājiem.

## 4.attēls. Datu plusmu diagramma lietotājiem.

## 4.3.3. Datu plusmu diagramma administratoram.

## 

## 5.attēls. Datu plusmu diagramma administratoram.

## 4.4. Aktivitāšu diagramma (Activity)

## Šajā sadaļā tiks izveidota aktivitāšu diagramma, kas vizualizēs sistēmas darbību un procesus, norādot to secību un izpildes loģiku.

## 4.4.1. Aktivitāšu diagramma - 1.daļa

## 

## 6.attēls. Aktivitāšu diagramma - 1.daļa

## 4.4.2. Aktivitāšu diagramma - 2.daļa

## 

**7.attēls. Aktivitāšu diagramma - 2.daļā**

## 

## 

## 4.5. Lietojumgadījumu diagramma (Use Case)

## Šajā sadaļā tiks izstrādāta lietojumgadījumu diagramma, kas identificēs un aprakstīs sistēmas funkcijas un to mijiedarbību ar lietotājiem.

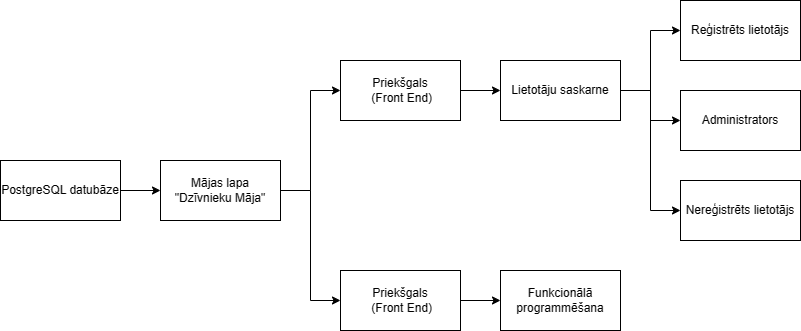
## 

## 8.attels. Lietojumgadījumu diagramma (Use Case)

## 4.6. Sistēmas moduļu apraksts un algoritmu shēmas

Mājas lapai “Dzīvnieku māja” sastāv no priekšpuses (Front End) un no aizmugures (Back End). (Skat. 9.attēlu - **Sistēmas moduļu diagramma**)

Priekšgals ir tīmekļa izstrādes daļa, kas ir atbildīga par lietotāja interfeisa izveidi, kas mijiedarbojas ar lietotāju, izmantojot pārlūkprogrammu. Tā “Dzīvnieku māja” projektā tiek izmantotas tādas programmēšanas valodas kā CSS un JavaScript, lai izveidotu mājas lapa, vizuālo dizainu un funkcionalitāti.

Aizmugure ir daļa no tīmekļa izstrādes, kas ir atbildīga par datu apstrādi un glabāšanu, kā arī mijiedarbību ar tīmekļa lietojumprogrammas klienta pusi, izmantojot API. Priekš sava projekta es izmantoju sekojošas programmēšanas valodas, kā Java, PostgreSQL, lai izveidot mājas lapas aizmuguri.

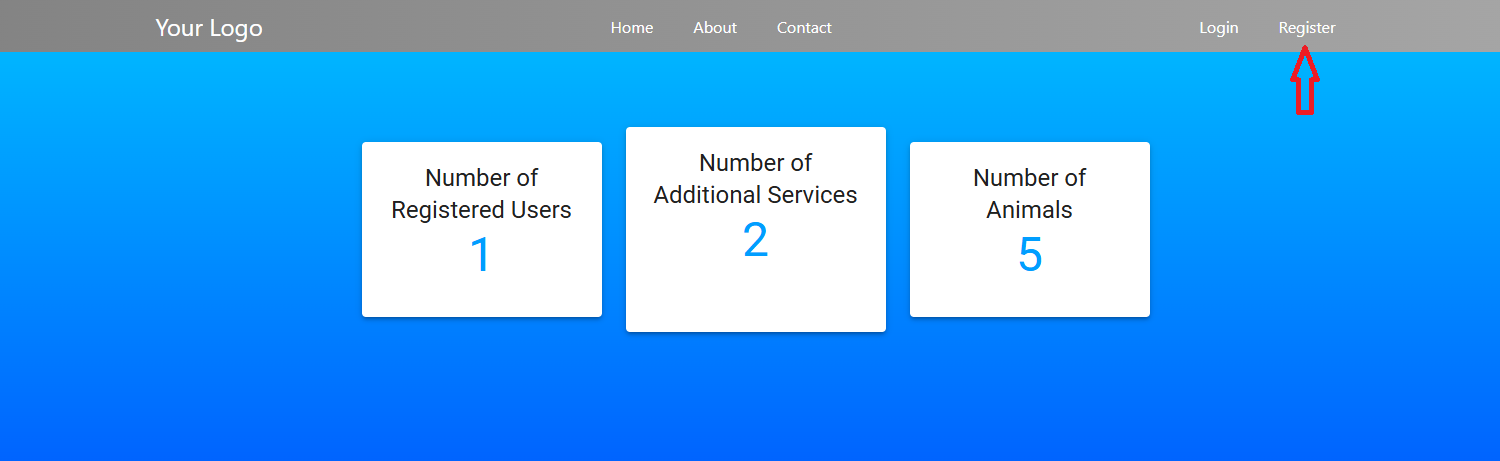
**9.attēls. Sistēmas moduļu diagramma**

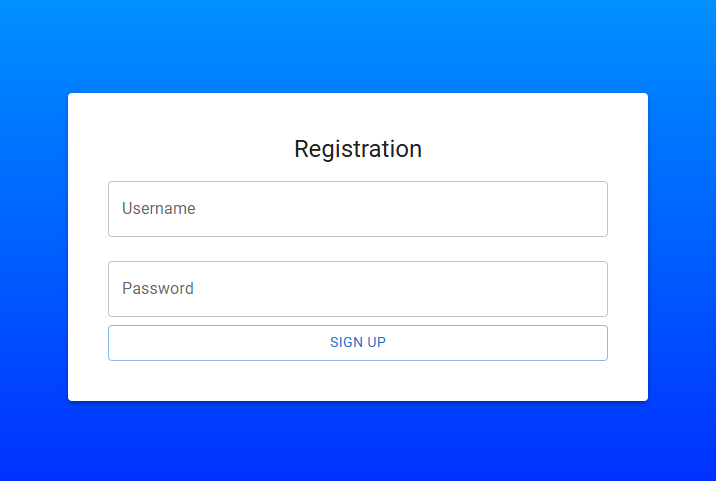
# 5. Lietotāju ceļvedis

## Šajā nodaļā tiks paradīts un pateikts solis pa solim, kā lietotājs var tikt kāda mājas lapas sadaļā un izmantot tas sadaļas funkcijas.

## 5.1. Mājas lapas reģistrācijas sadaļas izmantošana

Atverot lapu, lietotājs redzēs mājaslapas galveno lapu, kurā viņš var nospiest pogu, “Register”(Skat. 10.attēlu), nospiežot kuru lietotājs tiks aizvadīts uz lietotāja reģistrēšanas formu, kurā lietotājam jāievada lietotāja dati, tādi ka, lietotājvārds un parole(Skat. 11.attēlu), lietotājs nevarēs piereģistrēties ar lietotājvārdu kurš jau ir aizņemts, veiksmīgi piereģistrējoties lietotājs tiks aizvadīts uz autorizācijas sadaļu.

**10.attēls. Reģistrācijas sadaļas izmantošana**

**11.attēls. Reģistrācijas sadaļas izmantošana**

## 

## 5.2. Mājas lapas autorizācijas sadaļas izmantošana

Izmantojot informāciju no iepriekšējās reģistrācijas sadaļas, mēs varam saprast, kā lietotājs pēc veiksmīgas reģistrācijas tiks aizvadīts uz autorizācijas sadaļu, arī lietotājs atrodoties uz galvenās lapas var nospiest pogu “Login”(Skat. 12.attēlu) un tiks virzīts uz autorizācijas sadaļu, šajā sadaļā lietotājam tiek dota iespēja autorizēt savu kontu, izmantojot reģistrācijas sadaļā ievadītos datus, piemēram, lietotājvārdu un paroli, un, ja dati atbilst iepriekš datubāzē ievadītajiem datiem, lietotājs tiek novirzīts uz galveno lapu, bet jā lietotājs ievadījis nepareizos datus, viņam tiks paradīts paziņojums par to kā dati nav ievadīti pareizi, un lietotājam būs iespēja vēlreiz ievadīt datus. (Skat. 13.attēlu)

**12.attēls. Autorizācijas sadaļas izmantošana**

## 

## 13.attēls. Autorizācijas sadaļas izmantošana

## 5.3. Mājas lapas dzīvnieku tabulas izmantošana un apskatīšana

Lietotājam ir iespēja tikt uz dzīvnieku tabulu ja viņš ir veiksmīgi autorizējies mājaslapā, galvenā lapā paradīsies poga “Table with animals”(Skat. 14.attēlu), nospiežot kuru, lietotājs tiks novirzīts uz tabulu ar dzīvnieku, kurā lietotājs var apskatīties visus pieejamus dzīvniekus(Skat. 15.attēlu).

**14.attēls. Dzīvnieku tabulas apskatīšana**

# 

# 

# 15.attēls. Dzīvnieku tabulas apskatīšana

# 5.4. Dzīvnieku pievienošana

# Atrodoties sadaļā ar dzīvnieku tabulu, administrators vai moderators var nospiest pogu "Create" (Skat. 16.attēlu), un tad viņam būs pieejama dzīvnieka pievienošanas forma, lai pievienotu dzīvnieku, ir jāaizpilda visi dati, piemēram, vārds, dzīvnieka veids (suns, kaķis utt.), dzīvnieka dzimšanas datums un dzimums, un pēc tam nospiest pogu "Save"(Skat. 17.attēlu).

# 

# 16.attēls. Dzīvnieku pievienošana

# 

## 

**17.attēls. Dzīvnieku pievienošanas forma**

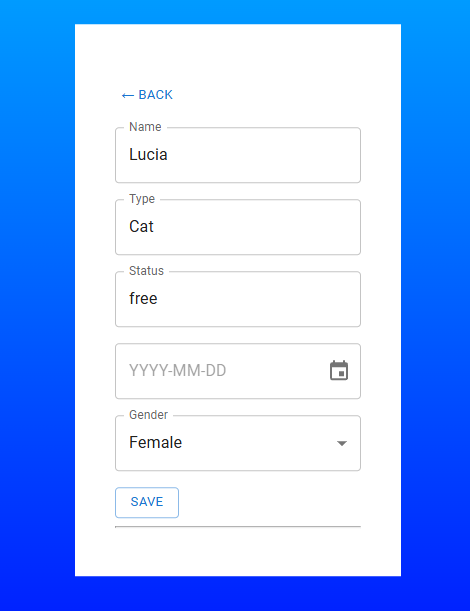
## 5.5. Esoša dzīvnieka datu rediģēšana

Atrodoties sadaļā ar dzīvnieku tabulu, administrators vai moderators var noklikšķināt uz pogas "Edit" rindā ar dzīvnieku, kura datus viņš vēlas mainīt (Skat. 18.attēlu) , tad atvērsies dzīvnieka datu rediģēšanas forma, lai mainītu dzīvnieka datus, ir jāaizpilda lauki kuros ir nepieciešams izmainīt datus, piemēram, vārds, dzīvnieka veids, dzimšanas datums vai dzimums, tad noklikšķiniet uz pogas "Save" (Skat. 19.attēlu).

# 

# 

**18.attēls. Dzīvnieka datu rediģēšana**

****

**19.attēls. Dzīvnieka rediģēšanas forma**

# 

# 5.6. Dzīvnieku bildes pievienošana

# Atrodoties sadaļā ar dzīvnieku tabulu, administrators vai moderators var noklikšķināt uz pogas "Images" rindā ar dzīvnieku, kuram viņš vēlas pievienot bildi (Skat. 20.attēlu) , tad atvērsies dzīvnieka bildes pievienošanas logs, kurā administrators vai moderators var pievienot bildes dzīvniekam izvelējoties tas no sava datora.(Skat. 21.attēlu).

# 20.attēls. Dzīvnieku bildes pievienošana

# 

**21.attēls. Dzīvnieku bildes pievienošana**

# 

# 5.7. Lietotāju tabulas apskatīšana

A screenshot of a computer

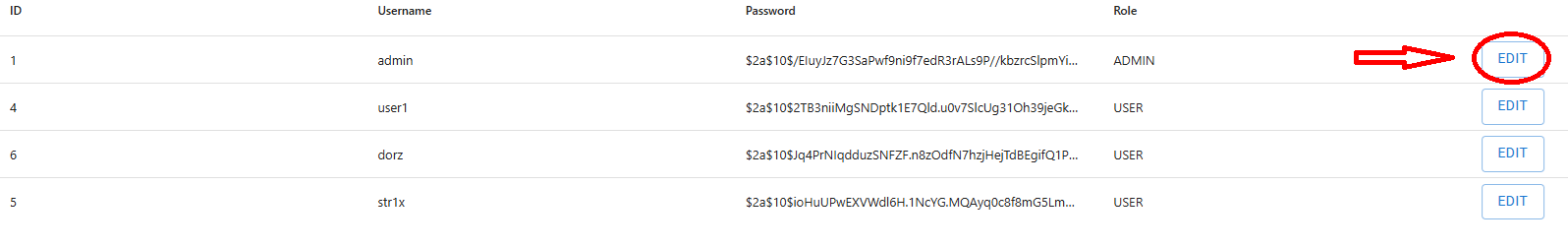
Description automatically generatedAtrodoties mājaslapas galvenajā lapā, administratoram ir iespēja piekļūst lietotāju tabulai, kur viņš var apskatīt visus mājaslapas lietotājus, kā arī veikt vajadzīgas darbības, tādas kā, izveidot jaunu lietotāju vai rediģēt lietotāja datus, noklikšķināšot uz pogas “Table with users”(Skat. 22.attēlu) , administrators tiek aizvadīts uz lietotāju tabulu (Skat. 23.attēlu).

**A screenshot of a computer

Description automatically generated22.attēls. Lietotāju tabulas apskatīšana**

**23.attēls. Lietotāju tabulas apskatīšana**

# 5.8. Lietotāja datu rediģēšana

Atrodoties lietotāju tabula, administratoram ir iespēja rediģēt vajadzīgos datus jebkuram lietotājam, nospiežot pogu “Edit” (Skat. 24.attēlu) kas atvērs lietotāja datu rediģēšanas formu, kurā administrators var izmainīt datus un saglabāt tos nospiežot pogu “Save” (Skat. 25.attēlu).

**24.attēls Lietotāja datu rediģēšana**

**A screenshot of a login screen

Description automatically generated**

**25.attēls Lietotāja datu rediģēšana**

# 5.9. Jauna lietotāja izveidošana

A close up of a text

Description automatically generatedAtrodoties lietotāju tabulā, administratoram ir iespēja izveidot jauno lietotāju, nospiežot pogu “Create”(Skat. 26.attēlu) kas aizvadīs viņu uz lietotāja izveidošanas formu, kur administrators aizpildot visus laukus tādus, kā lietotājvārds, parole un loma, var izveidot jaunu lietotāju un saglabāt to nospiežot pogu “Save”(Skat. 27.attēlu).

A screenshot of a login screen

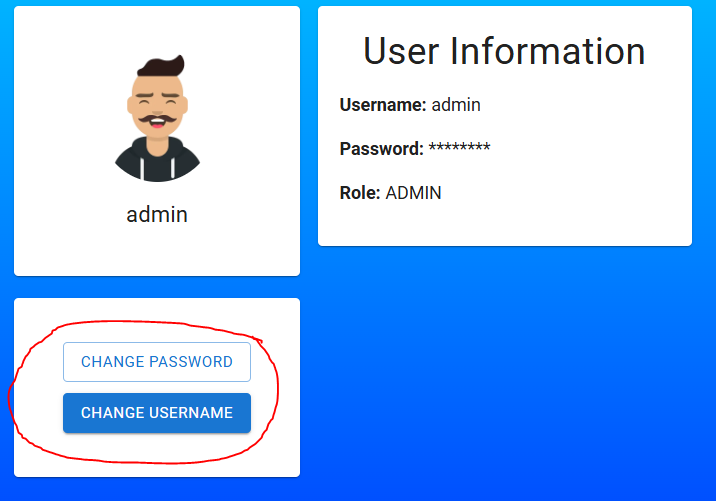
Description automatically generated**26.attēls Jauna lietotāja izveidošana**

**27.attēls Jauna lietotāja izveidošana**

# 5.10. Profila atveršana

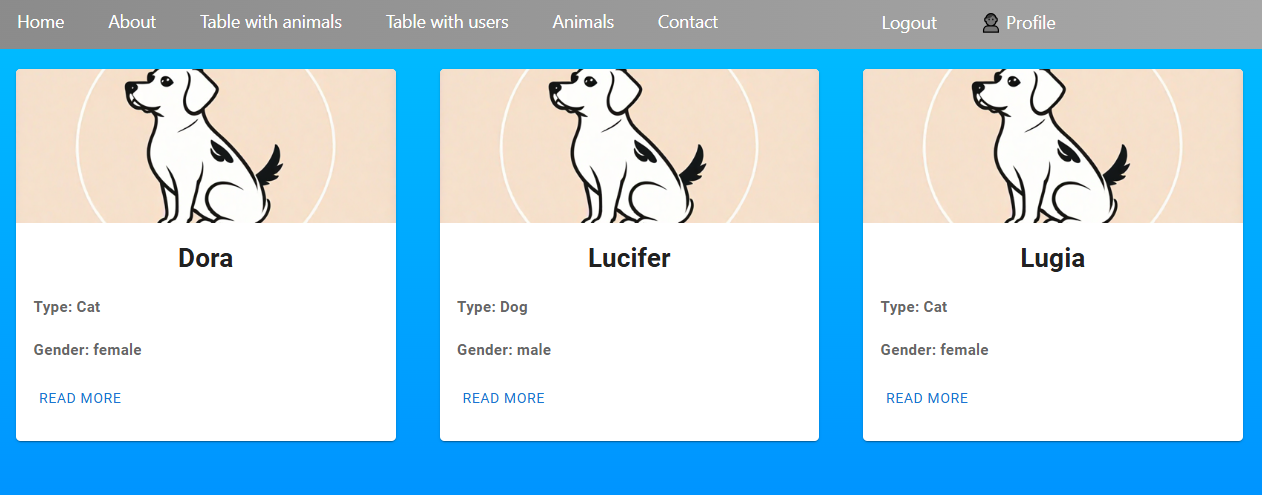
Atrodoties hebkurā lapā lietotājiem ir iespēja atvert savu profilu, nospiežot pogu “Profile”(Skat. 27.attēlu), kurā lietotājam bus iespēja vai nu izmainīt paroli, vai izmainīt lietotājvardu, nospiežot izvelēto pogu.(Skat. 28.attēlu)

**27.attēls Profila atveršana**



**28.attēls Profila atveršana**

**5.11. Dzīvnieka katalogu apskatīšana**

Atrodoties jebkurā lapā autorizētam lietotājam ir iespēja apskatīties dzīvnieka katalogu, nospiežot pogu “Animals”(Skat. 29.attēlu), katalogā lietotājs var apskatīt dzīvnieka detalizēto informaciju nospiežot pogu “Read more” (Skat. 30.attēlu).

**29.attēls Dzīvnieka katalogu apskatīšana**

**30.attēls Dzīvnieka katalogu apskatīšana**

# 6. Testēšanas dokumentācija

Šajā nodaļā tiks aprakstīts detalizēts programmatūras testēšanas process, kas nodrošina augstu kvalitāti un pareizu darbību. Testēšanas dokumentācija ir būtiska, lai veiktu gan manuālo, gan automatizēto testēšanu, ietverot tādus testēšanas veidus kā white box un black box.

Manuālā testēšana ietver manuālas pārbaudes un interaktīvu mājaslapas testēšanu, lai nodrošinātu lietotājam draudzīgu interfeisu un funkcionalitāti. Automatizētā testēšana tiek veikta, izmantojot iepriekš izstrādātus testa scenārijus, kas automātiski pārbauda sistēmas stabilitāti un pareizu darbību atkārtojamos apstākļos.

Svarīga ir arī testēšanas dokumentācija, kur detalizēti aprakstīti visi testi, to rezultāti un atbilstība sistēmas specifikācijām. Šie dati tiek reģistrēti Excel tabulā, kas sniedz skaidru pārskatu par testēšanas procesu un tā rezultātiem.

Turklāt dokumentācijā būs iekļauts detalizēts apraksts par izmantotajiem testēšanas veidiem, piemēram, white box un black box testēšanu. White box testēšana koncentrējas uz programmas iekšējo struktūru un koda pārbaudi, lai nodrošinātu tā pareizu darbību un efektivitāti. Black box testēšana fokusējas uz programmas ārējo interfeisu un funkcionalitāti, pārbaudot tās darbību no lietotāja skatpunkta.

Šāda dokumentācija ir būtiska, lai nodrošinātu sistēmas atbilstību specifikācijām, uzticamību un darbības stabilitāti. Tā sniedz ieskatu par veiktajiem testiem, to rezultātiem un kalpo par pamatu turpmākai sistēmas uzlabošanai.

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

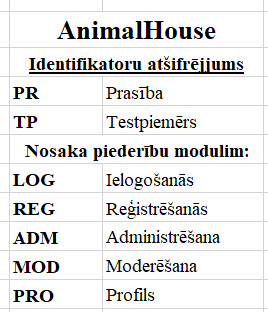
Testēšanas procesā tiks plaši izmantotas gan white box, gan black box metodes. Šīs pieejas nodrošinās visaptverošu sistēmas pārbaudi, piedāvājot dažādus skatupunktus uz tās darbību un funkcionalitāti.

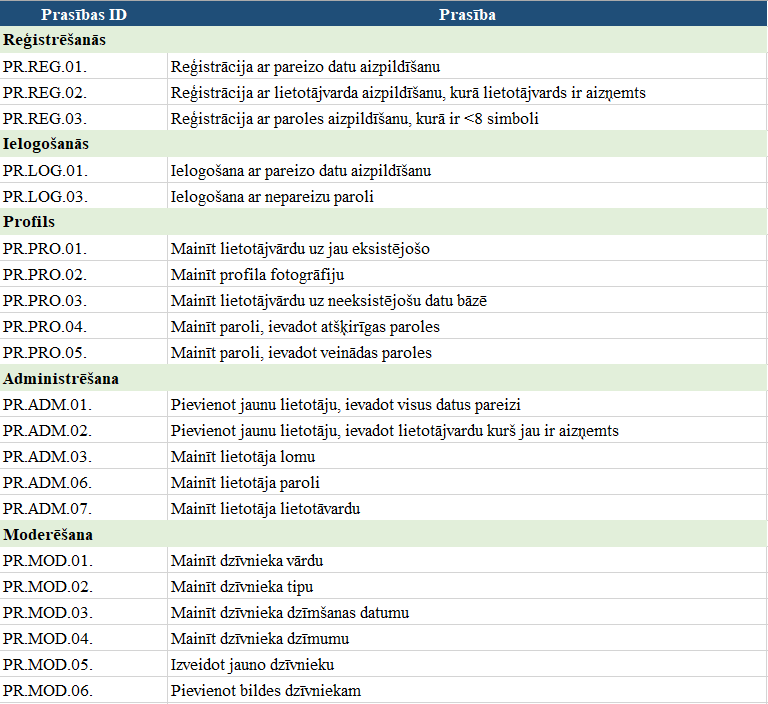
White box testēšana sniegs ieskatu sistēmas iekšējā struktūrā un kodā. Tās galvenais mērķis ir izpētīt programmas loģiku un darbību, identificēt iespējamos defektus un optimizēt kodu. Šī metode palīdzēs nodrošināt efektivitāti un drošību, garantējot, ka programmatūra darbojas pareizi un ir pasargāta no drošības riskiem.

Savukārt black box testēšana koncentrēsies uz sistēmas uzvedību no gala lietotāja skatupunkta. Šajā metodē testētāji pārbaudīs sistēmas funkcionalitāti un saskarni, neņemot vērā tās iekšējo struktūru. Galvenais mērķis ir nodrošināt, ka programmatūra atbilst specifikācijām un piedāvā lietotājiem kvalitatīvu un intuitīvu pieredzi.

Testu rezultātu reģistrēšanai un organizēšanai tiks izmantota Excel tabula. Tā nodrošinās skaidru struktūru un vieglu sapratni gan testētājiem, gan izstrādātājiem. Tabulā tiks ierakstīti visi veiktie testi, to rezultāti un atbilstība specifikācijām, kas nodrošinās pārskatāmu un efektīvu testēšanas procesu.

Šādas pieejas kombinācija nodrošinās pilnīgu pārbaudi un augstu programmatūras kvalitāti. White box un black box metodes ļaus veikt visaptverošu sistēmas pārbaudi, savukārt Excel tabula nodrošinās efektīvu rezultātu dokumentēšanu un pārskatāmību. Tādējādi tiks optimizēts gan testēšanas process, gan rezultātu dokumentācija, nodrošinot augstu programmatūras kvalitāti un uzticamību.

**28.attēls Rīku apraksts**



**29.attēls Prasības apraksts**

## 6.2. Testpiemēru kopa



## 

## 

## 

## 

## 6.3. Testēšanas žurnāls



# 8. Secinājumi

#### Sasniegums, rezultāta novērtējums

Mājaslapa "AnimalHouse" veiksmīgi nodrošina lietotājiem iespēju pirkt vai pardot dzīvniekus tiešsaistē, piedāvājot vienkāršu un intuitīvu lietotāja interfeisu.

**Izvirzīto uzdevumu sasniegšanas analīze**

##### Kas izdevies:

* **Lietotāja interfeiss:** Dizains ir ērti lietojams, nodrošinot vieglu navigāciju.
* **Rezervācijas funkcionalitāte:** Sistēma nodrošina visas nepieciešamās funkcijas pīrkšanas vai pardošanas veikšanai, ieskaitot dzīvnieka izvēli.

##### Kas nē:

* **Papildus funkcionalitātes trūkums:** Daži plānotie uzlabojumi, piemēram, mājaslapas tūlkošana , vēl nav realizēti.

#### Darba apjoms

##### Sistēmas struktūra:

* **Failu skaits:** Aptuveni 40-50 faili.
* **Moduļu skaits:** 5 galvenie moduļi
* **Funkciju skaits:** Ap 20 funkcijas, kas nodrošina mājaslapas darbību.

##### Koda apjoms:

* **Pašrakstīto komandrindu skaits:**
  + Dizainam: 500 rindas apmēram.
  + Funkcionalitātei: 1500 rindas.
* **Kopīgais komandrindu skaits (ar ģenerētām un kopīgi veidotām):**
* Aptuveni 3000 rindas.

##### Testēšana

Programma tika testēta uz dažādiem pārlūkiem (Chrome, Microsoft Edge). Testēšana bija sekmīga, un tika novērstas visas atrastās kļūdas.

#### Problēmas un izaicinājumi

Izstrādes laikā bija vairākas problēmas:

* **Laika ierobežojums:** Liels izaicinājums bija tas, ka bīja viss saplānots izveidot mājaslapu "AnimalHouse" lidz eksāmena dienai, bet tika nosacīti nodošanas termiņš samazināts.

#### Nākotnes ieceres

* **Uzlabot integrāciju:** Pievienot iespēju ar mēneš maksājumu

# 9. Lietoto terminu un saīsinājumu skaidrojumi

|  |  |
| --- | --- |
| **Termins** | **Skaidrojums** |
| **ID** | **Unikāls identifikators** |
| **Dropdown** | **nolaižamā izvēlne** |
| **Date picker** | **datuma izvēlne** |
| **Backend** | **aizmugures daļa (vai servera puse)** |
| **Frontend** | **priekšējā daļa (vai lietotāja saskarne)** |
| **WhiteBox** | **Testēšanas metode, kur testētājam ir pilna informācija par sistēmas kodu un darbību.** |
| **BlackBox** | **Testēšanas metode, kur testētājam nav informācijas par sistēmas kodu un struktūru.** |
| **API** | **Lietojumprogrammu saskarne (Application Programming Interface)** |

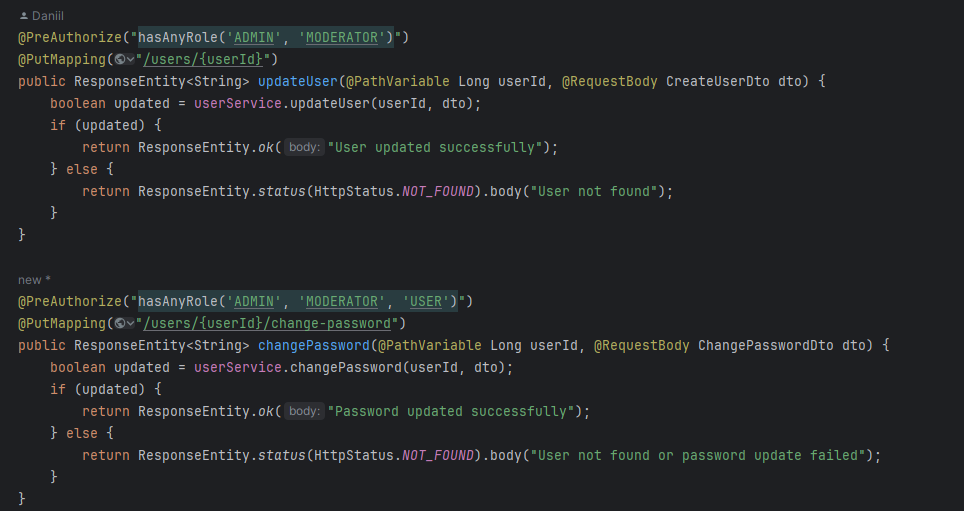
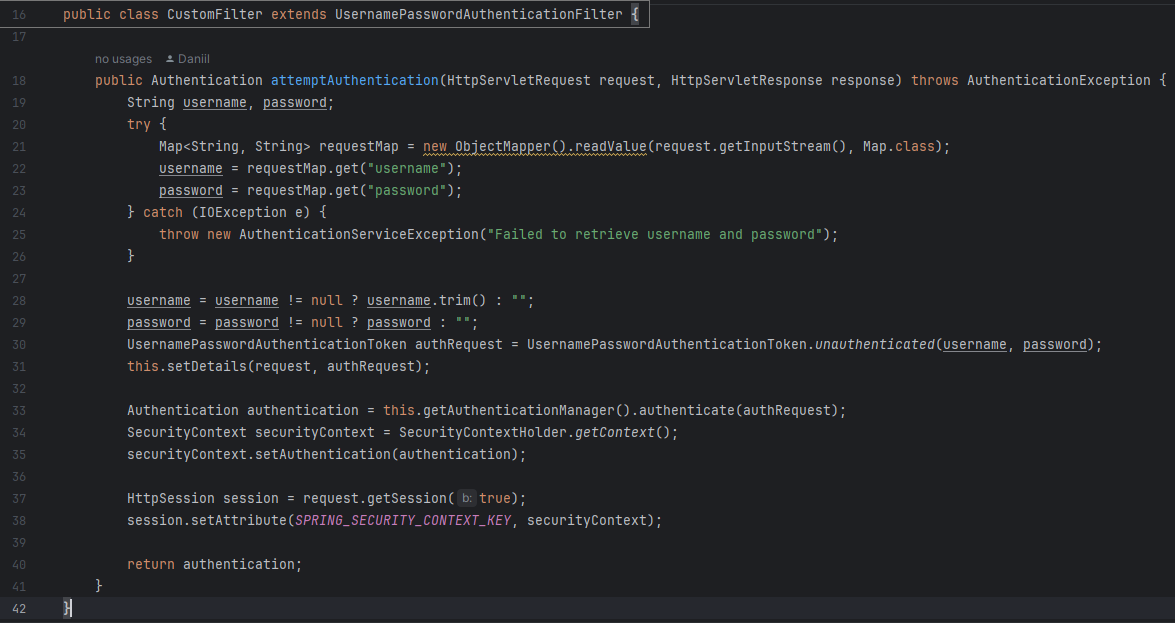
# 10. Literatūras un informācijas avotu saraksts

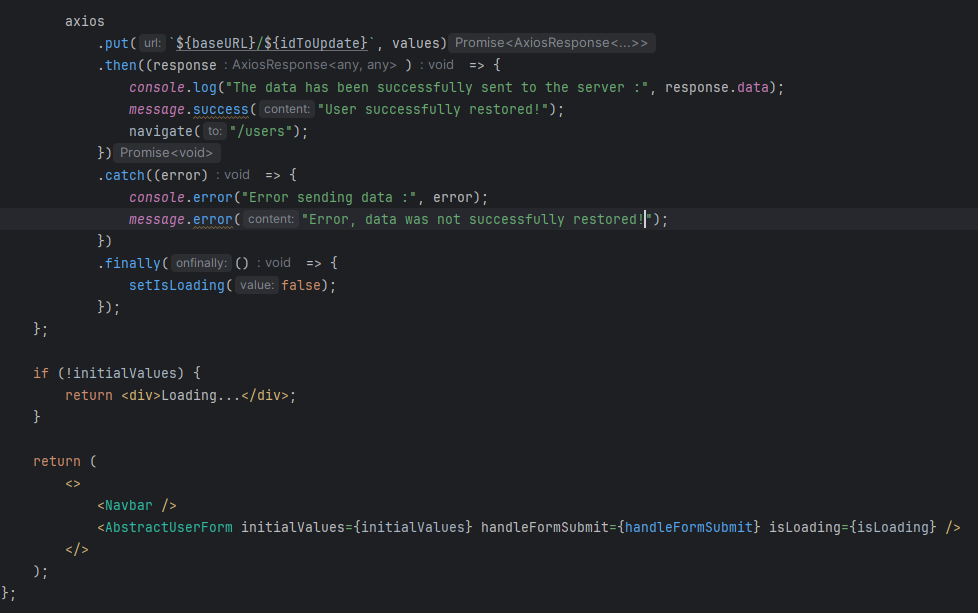
<https://www.npmjs.com/package/>

[React Tutorial (w3schools.com)](https://www.w3schools.com/react/)

<https://mdbootstrap.com/docs/react/>

# PIELIKUMS

**1.Pielikums**

**2.Pielikums**

Izstrādātās sistēmas pirmkods pieejams versionēšanas sistēmā Github, šeit:

[ImpulseLv/DT\_Eksamens\_DM: Dzīvnieku māja majaslapa (github.com)](https://github.com/ImpulseLv/DT_Eksamens_DM)

“AnimalHouse” mājas lapa pieejama, šeit:

[React App](http://104.248.129.29/)