

Rēzeknes Tehnoloģiju akadēmija

Inženieru fakultāte

Studiju virziens „Informācijas tehnoloģija, datortehnika, elektronika,
telekomunikācijas, datorvadība un datorzinātne”

SKOLU ĒDINĀŠANAS MONITORINGA SISTĒMA

KVALIFIKĀCIJAS DARBS

Autors

Haralds GLAUDĀNS

1. līmeņa profesionālās
augstākās izglītības
studiju programmas
“Programmēšana un datortīklu
administrēšana” 2. kursa students
Stud. apl. nr. hg19001
hg19001@edu.rta.lv

IT virziena vadītājs

Dr.sc.ing. Pēteris GRABUSTS

Zinātniskais vadītājs

Dr.sc.ing. Sergejs KODORS

Recenzents

Rēzekne

2022

ANOTĀCIJA

Kvalifikācijas darba tēma: Skolu ēdināšanas monitoringa sistēma.

Autors: Haralds GLAUDĀNS

Darba mērķis: izstrādāt tīmekļa vietni, kas veiks skolas atkritumu monitoringu.

Apraksts: šīs sistēmas nolūks ir padarīt skolu ēdināšanas sistēmas analīzi vienkāršāku, lai vienā vietā varētu savākt visu informāciju par apēsto porciju daudzumu, kas nākotnē varētu mazināt atkritumu daudzumu un veidot jaunas ēdienkartes pēc skolēnu gaumes.

Darba rezultāts: Tika izstrādāts gatavs programmatūras produkts - *Skolu ēdināšanas monitoringa sistēma*. Tāpat tika sagatavotas šī produkta lietošanas un instalācijas instrukcijas. Sagatavota video demonstrācija, kurā parāda, kā izstrādātā lietotne strādā.

Tehnoloģijas:

Python

Django

Mysql

Bootstrap

Nginx

Gunicorn

Github

Git

DB Management System

Darba saturs: 38 lappuses, 30 attēli, 54 literatūras avoti.

Atslēgas vārdi: ēdināšana, tīmekļa lietotne, Python, monitoringa sistēma.

ABSTRACT

Qualification Work Theme: The school catering monitoring system.

Author: Haralds GLAUDĀNS

Goal of the work: develop a website that will carry out school waste monitoring.

Description: the aim of this system is to make the analysis of the school catering system simpler so that all information on the amount of portions eaten can be collected in one place, which could reduce waste in the future and create new menus for the taste of the pupils.

Result: A ready-made software was developed - *The school catering monitoring system*. Instructions for use and installation of this product were also prepared. A video demonstration has been prepared showing how the developed application works.

Technologies:

Python

Django

Mysql

Bootstrap

Nginx

Gunicorn

Github

Git

DB Management System

The work contains 38 pages, 30 picture, 54 references.

Keywords: catering, Python, web application, monitoring system.

Apzīmējumu saraksts

1. IT – Informāciju tehnoloģijas;
2. “Skolu ēdināšanas monitoringa sistēma” – izstrādājamā tīmekļa lietotne;
3. Visual Studio Code – programma, kuru izmanto dažādu programmu izstrādē, kodu rakstīšanā;
4. Django – piemērota sarežģītu vietņu un tīmekļa lietojumprogrammu izstrādei, kas rakstīta programmēšanas valodā Python;
5. Python - populāra programmēšanas valoda, kura tika izmantota aplikācijas izstrādē;
6. Id - identifikators , piemērām, kārtas numurs.

Saturs

Ievads	6
1. Esošās sistēmas analīze	7
1.1 Atkritumu monitoringa sistēma <i>Winnow Sense</i>	8
1.2 Ēdināšanas pakalpojumu programmatūra <i>MarketMan</i>	9
1.3. Ēdināšanās programmatūras nodrošinātājs <i>Jamix</i>	10
2. Sistēmas apraksts	11
3. Izstrādes tehnoloģiju izvēle.....	15
4. Izstrādes procesa apraksts	16
5. Instalācijas izstrāde	27
5.1 Virtuālas vides instalācija.....	27
5.2 Django ietvara instalācija.....	29
5.2 Projekta koda augšupielāde.....	32
5.3 Projekta pārņemšana	32
5.4 Darbs ar datu bāzi	32
Secinājumi un priekšlikumi.....	33
Kopsavilkums.....	34
Summary	35
Literatūra	36

Ievads

Aktualitāte: mūsdienās informācijas tehnoloģijas (IT) sāk pārņemt ikvienu dzīves jomu, gan darbos, gan studijās, gan izklaidē. Praktiski jebkuram iedzīvotājam pa rokai ir kāda no viedajām ierīcēm, vai nu tas ir viedtālrunis, vai planšete. Šīs, kā arī datortehnika – portatīvie un stacionārie datori galda datori – ikvienā jomā tā tiek pielietota ar vien vairāk un vairāk. Skolas nav izņēmums, ar katru gadu skolas paliek digitalizētas, bet ēdināšanas monitoringa sistēma stāv uz vietas un nekā neprogresē, neiet kopā ar laiku, ar tehnoloģijām, tāpēc tika izvēlēta tēma – “Skolu ēdināšanas monitoringa sistēma”. Šobrīd skolās vienīga iespēja savākt statistiku par apēstām porcijām un atkritumu daudzumu ir izmantojot *Excel*, bet tam ir trūkumi, kas palēlina visu skolu datu savākšanu vienā vietā un apgrūtina ātro datu pārskatu.

Pārtikas atkritumu (*food waste*) ir starptautiska problēma, cilvēki gada laikā izmet ārā 931 miljonu tonnu pārtikas atkritumu - tas ir aptuveni 23 miljoni kravas automašīnu, ko var septiņas reizes apbraukt apkārt Zemei, kā arī pārtikas atkritumi rada 8-10% siltumnīcefekta gāzu emisiju, kas nodara ne tikai cilvēkiem kaites, bet arī zemei.

Darba mērķis: izstrādāt tīmekļa vietni, kas veiks skolas ēdināšanas monitoringu.

Uzdevumi:

1. Analizēt esošu situāciju;
2. Paskaidrot izstrādes tehnoloģijas izvēli;
3. Aprakstīt kodēšanas procesu;
4. Izstrādāt lietotāja instrukciju;
5. Izstrādāt lietotnes instalāciju;
6. Veidot secinājumus un priekšlikumus

Rezultāti:

Izstrādāto tīmekļa lietotni var apskatīt pēc adreses - <https://www.kittcc.net/>

1. Esošās sistēmas analīze

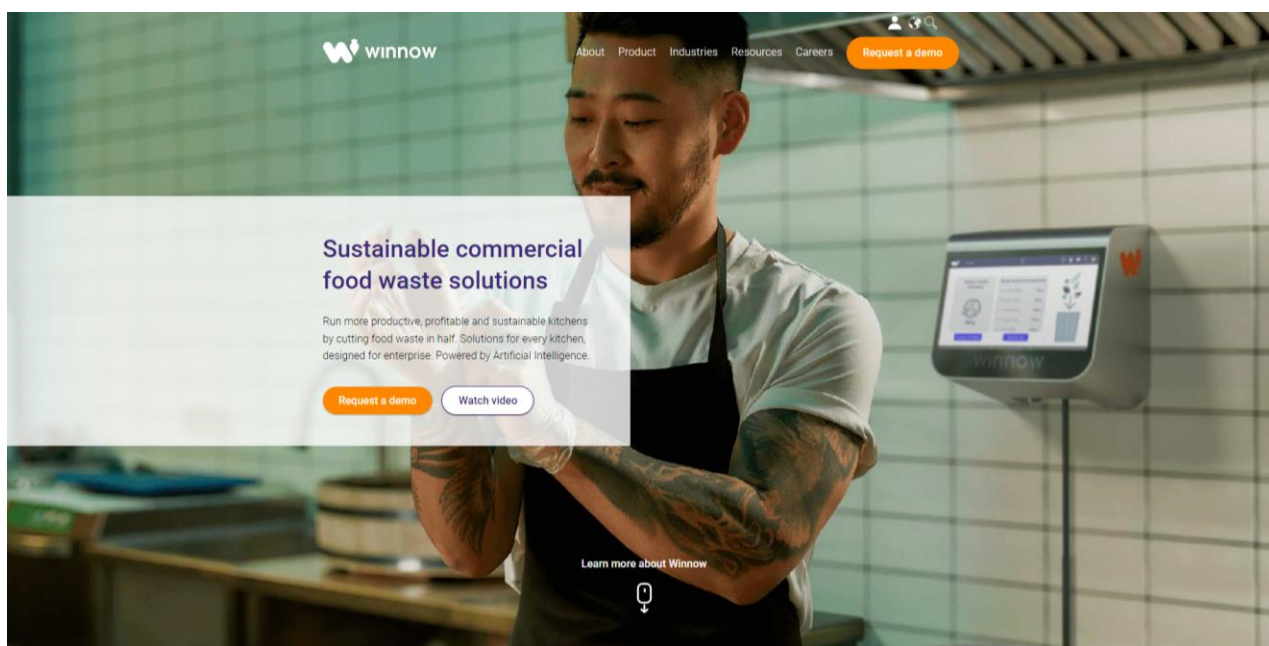
Pirms Skolu ēdināšanas monitoringa sistēmas izstrādes tika veikta līdzīgu sistēmu analīze – tika izsecināts, ka tīmekļa vietnēs šādu sistēmu, kas būtu saistīti ar skolu nav, bet ir programma *Excel*, kurai ir trūkumi salīdzinot ar autora sistēmu.

Excel ir trūkumi, kas palēlina visu skolu datu savākšanu vienā vietā un apgrūtina ātro datu pārskatu, jo datu pārsūtīšana notiek caur ē-past sistēmu un katrai skolai ir atsevišķs, kas nav labi un ātri pārskatāms priekš pašvaldības, vai, piemēram, ministrijas.

Autora izstrādātājā tīmekļa vietnē šāda problēma tika atrasināta, jo vairāk pašvaldībām nav jāmeklē *Exceli* datnes, tagad visas skolas ir vienā datubāzē - Pašvaldībām ir pieeja pie savu skolu loka, bet profils “Valdība” redz visas skolas Latvijā. Pirms izstrādes tika apskatītas līdzīgas sistēmas, tādās ka Winnow Sense, MarketMan un Jamix.

1.1 Atkritumu monitoringa sistēma *Winnow Sense*

Winnow Sense [22] (skat. 1.1 att.) ir atkritumu monitoringa sistēma, kas ļauj iegūt plašāku informāciju par klientu, izmesto atkritumu daudzumu, ēdienkartes izvēli, porciju kontroli, ir pieejama kā stacionārā iekārta, kas tiek piemontēta pie sienas ar urnu lejā. Šīs sistēmas funkcionalitāti var apskatīt video, kas ir izlikts oficiālajā *Youtube* kontā, ar *video demo* [23].



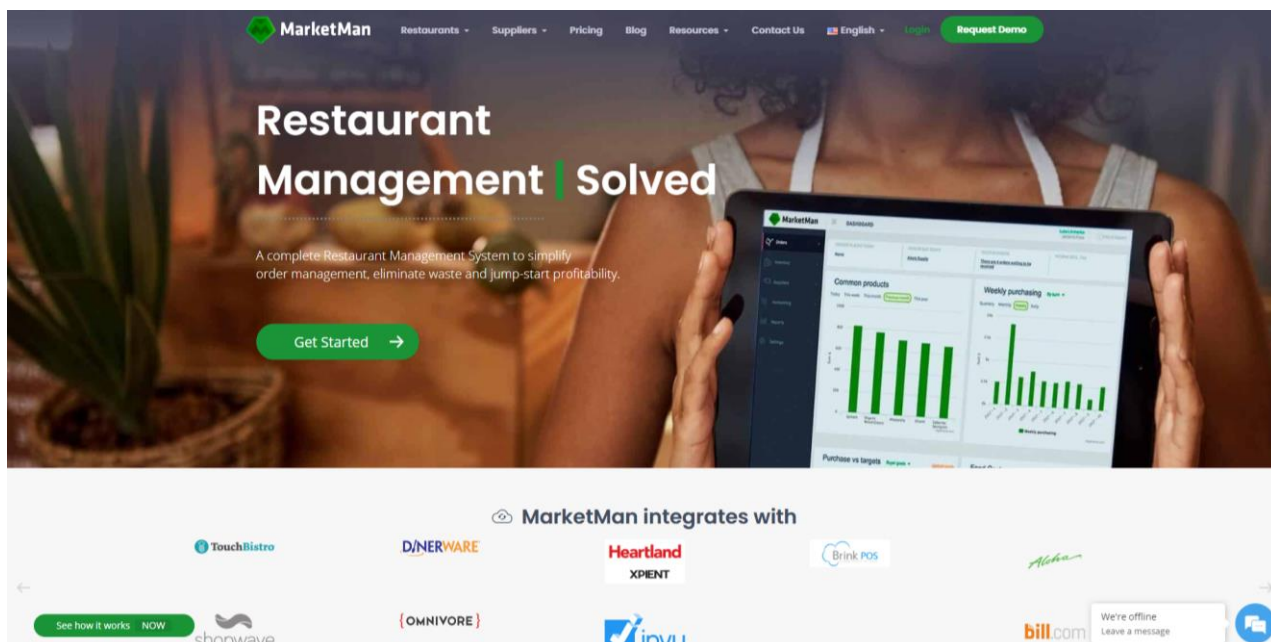
1.1 attēls. *Winnow Sense* tīmekļa lietotne

Pamatā sistēma tiek piedāvāta šāda funkcionalitāte:

- **Bez kontakta atkritumu svara un attēlu ierakstīšana** – izmetot atkritumus urnā, sistēmā ar svaru palīdzību noskaidro cik lielu atkritumu daudzumu izmeta klients un izvada uz monitora atkritumu svaru.
- **Precīzi noskaidro, ko klients izmeta urnā** – izmestos atkritumus sistēma nofotografē un atpazīst, kas tas ir par produktu un uz ekrānu izvada produkta fotogrāfiju un svaru, kā arī tīmekļa lietotnē var apskatīt katra produkta atkritumu daudzumu.
- **Nosaka klientu tendences laika gaitā** – tīmekļa lietotnē ir iespēja apskatīt produktu nedēļas topu, cik un kāds produkts tika izmests urnā.

1.2 Ēdināšanas pakalpojumu programmatūra *MarketMan*

Marketman [24] (skat. 1.2 att.) ir ēdināšanas pakalpojumu operatoru un to piegādātāju sadarbības programmatūra. Šīs sistēmas funkcionalitāti var apskatīt video, kas ir izlikts oficiālajā *Youtube* kontā, ar video piemēru [25].



1.2. attēls *MarketMan* tīmekļa lietotne

Pamatā sistēma tiek piedāvāta šāda funkcionalitāte:

- **Atkritumu atskaite** - atkritumu izsekošanas iespējas palīdz restorānu darbiniekiem palikt proaktīviem un samazināt pārtikas izmaksas, uzlabojot redzamību, no kurienes nāk atkritumi. Analizētu un saprastu, no kurienes rodas atkritumi, kā tiem ir tendence un kā ir saistīti jūsu pirkumi un atkritumi.
- **Atbilstoši porciju ēdieni** - viena no MarketMan pavārgrāmatu funkcijas galvenajām funkcijām ietver precīzu ēdienkartes plānošanu. Šī funkcija ļauj restorāniem gūt labāku ieskatu par izmaksām. Viņiem būs arī vieglāk izsekot krājumiem un analizēt tādas lietas kā, piemēram, kuras preces dod vislielāko peļņu, kuras preces ir jākorrigē un kuras jāizņem no izvēlnes.
- **Krājumu pārvaldnieks** - runa ir par krājumu uzskaiti. Šie risinājumi palīdz izstrādāt stratēģiju, kuras preces pasūtīt, lai samazinātu atkritumu daudzumu, vienlaikus ļaujot jums kontrolēt savu budžetu un izdevumus. Tie var sniegt praktiskus datus par jūsu finansiālo darbību. Varat tos izmantot, lai aprēķinātu pārtikas izmaksu procentus un samazinātu pārmērīgu iepirkšanos, tāpēc ir mazāka iespēja, ka nedēļas beigās nāksies izmest neizmantoto pārtiku.

1.3. Ēdināšanās programmatūras nodrošinātājs *Jamix*

Jamix [26] (skat. 1.3 att.) ir virtuves vai ēdināšanas pakalpojumu pārvaldības programmatūras nodrošinātājs. Šīs sistēmas funkcionalitāti var apskatīt video, kas ir izlikts oficiālajā *Youtube* kontā [27].



1.3. attēls Jamix tīmekļa lietotne

Pamatā sistēma tiek piedāvāta šāda funkcionalitāte:

- **Pārvaldīt neierobežotu skaitu recepšu sistēmā** – receptes ir ātri un viegli izveidojamas, atlasot sastāvdaļas no pilnajiem sastāvdaļu sarakstiem, kas iekļauti virtuves sistēmā. Pateicoties daudzpusīgajām grupēšanas iespējām, receptes ir viegli sakārtotas. Receptēs varat pievienot arī informāciju par diētām. Ja nepieciešams, instrukcijas var sadalīt vairākos posmos. Ja virtuves sistēmu izmantojat mobilajā ierīcē, varat ar to uzņemt attēlus un saglabāt tos tieši savā receptē.
- **Daudzpusīgas iespējas izvēlņu plānošanai un pārvaldībai** - ēdienkartes pārvaldība ir viena no galvenajām funkcijām tīmeklī balstītajā JAMIX Kitchen Intelligence sistēmā, kas visaptveroši atbalsta jebkura izmēra un veida ēdināšanas pakalpojumus.
- **Elektroniski produktu datu atjauninājumi no jūsu piegādātājiem** - būtiskā informācija par produktiem no piegādātājiem, tostarp cenas. Izmantojot pamatdatus, jūsu cenas un aprēķini vienmēr būs atjaunināti jūsu receptēs un ēdienkartēs visā sistēmā. Kad sistēmā ir precīza informācija par produktu, varat nosūtīt elektroniskus pasūtījumus saviem piegādātājiem.

2. Sistēmas apraksts

Sistēmas ideja radījās prakses laikā, izejot praksi Rēzeknes Tehnoloģiju akadēmijā, pārveidojot “Ementors” atkritumu prognozēšanas sistēmu no *Python* valodas uz *JavaScript* risinājumu. Vadītājs piedāvāja kvalifikācijas darba ietvaros izstrādāt tīmekļa lietotni, kas veiks atkritumu monitoringu dažādos vadības līmeņos.

Tīmekļa vietnes izstrādes sākumā, tika noteikts, izveidot tīmekļa lietotni *Django* ietvarā, kas īsteno ēdināšanas monitoringu skolās, kurā būtu triju veidu lomas: “skola”, “pašvaldība” un “valdība”. *Skola* var redzēt informāciju tikai par savu skolu un rediģēt ēdienkarti un skolēnu skaitu, un apēsto porciju skaitu. *Pašvaldība* spēj redzēt visas skolas sava novadā, un apskatīt informāciju par katru no tām, bet *valdība* redz visas skolas, kuras ir reģistrētas datubāzē.

Tīmekļa lietotne pēc paša iniciatīvas tika realizēta uz *Ubuntu* [45] servera, kas tika iznomāts pie *Digital Ocean* [29].

Lai piefiksētu sistēmas prasības pēc intervijas, autors sagatavoja sistēmas prasību specifikācijas (PPS) uzmetumu, noformējot to brīvā formā. Turpmāk tekstā var iepazīties ar PPS uzmetumu.

Tīmekļa lietotne „Skolu ēdināšanas monitoringa sistēma” (PPS)

Autors:

*programmēšana un datortīklu administrēšana, 2. kursa students,
Haralds Glaudāns*

1. Ievads

1.1. Nolūks

Šī dokumenta nolūks ir aprakstīt izstrādājamās tīmekļa vietnes idejas konceptus un prasības.

1.2. Darbības sfēra

izstrādājama sistēmā ir tīmekļa lietotne, kas ir orientēta uz skolas, pašvaldību, valdību un ēdināšanas sfēru.

1.3. Definīcijas, akronīmi un saīsinājumi

„Skolu ēdināšanas monitoringa sistēma” – izstrādājamā tīmekļa vietne; Django – tīmekļa lietotne.

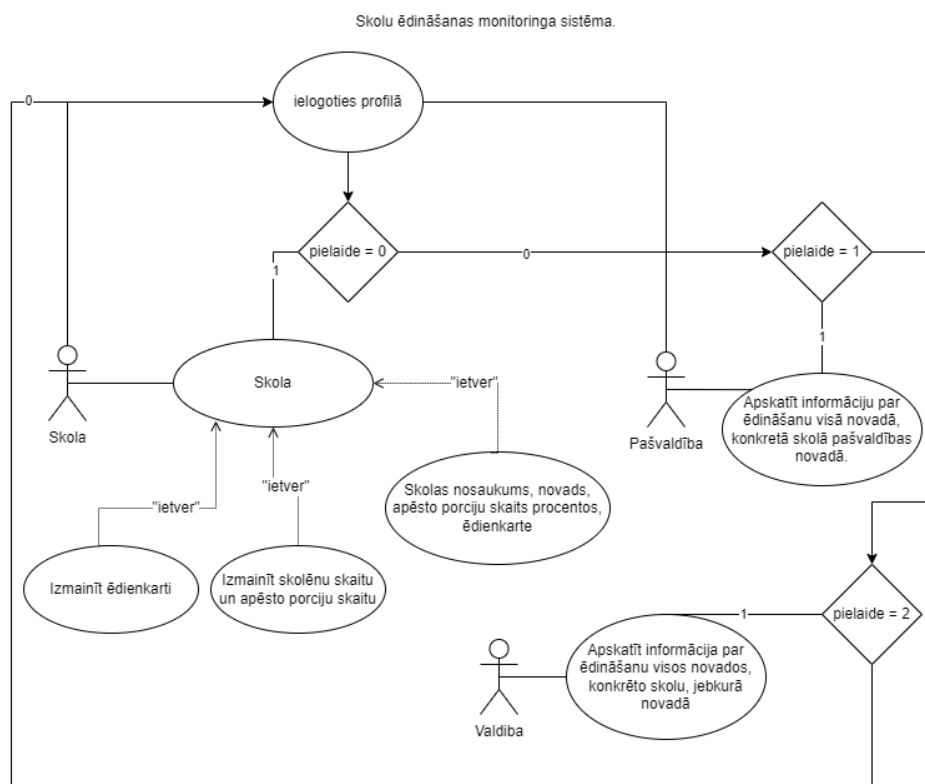
2. Vispārējais apraksts

2.1. Produkta perspektīva

Izstrādājamā tīmekļa lietotne ir paredzēta skolām, pašvaldībām un valdībai.

2.2. Produkta funkcijas

Tīmekļa lietotnē pastāv iespēja apskatīt statistiku par skolu apēsto porciju skaitu, ēdienkarti uz nedēļu un skolēnu skaitu. 2.1 attēla var redzēt sistēmas koncepcijas uzmetumu.



2.1. attēls. Uzmetums ar sistēmas koncepciju

2.3. Vispārējie ierobežojumi

2.3.1. Skolām nevar redzēt citu skolu statistiku.

2.3.2. Pašvaldības nevar redzēt citu pašvaldību skolu statistiku

2.3.3. Valdība var redzēt visu skolu statistiku

2.4. Pieņēmumi un atkarības

Ja lietotājs ir Skola tad pielaide ir vienāda ar 0, ja lietotājs ir pašvaldība tad pielaide ir vienāda ar 1, ja lietotājs ir valdība tad pielaide ir vienādā ar 2.

3. Specifiskās prasības

3.3. Funkcionālās prasības

3.3.2. Prasība – Lietotāju grupa Skola

3.3.2.1. ja lietotājs ir novada pārstāvis tad, pielaide ir vienāda ar 0, kas ļauj redzēt tikai savas skolas informāciju.

3.3.2.2. lietotājs var rediģēt skolas ēdienkarti, skolēnu skaitu un apēsto porciju skaitu.

3.3.3. Prasība – Lietotāju grupa Novads

3.3.3.1. Ja lietotājs ir novada pārstāvis tad, pielaide ir vienāda ar 1, kas ļauj redzēt visas skolas sava novada un apskatīt informāciju par tām.

3.3.4. Prasība – Lietotāju grupa Valdība

3.3.4.1. Ja lietotājs ir novada pārstāvis tad, pielaide ir vienāda ar 2, kas ļauj redzēt visas skolas sava novada un apskatīt informāciju par tām.

3.4. Projekta ierobežojumi

Sistēmu jārealizē *Python* valodā, pielietojot *Django* satvaru.

3. Izstrādes tehnoloģiju izvēle

Python 3 Build 4126. [11] — ir programmēšanas valoda, tas ir spēcīgs rīks programmu izveidei dažādiem mērķiem, kas ir pieejams pat iesācējiem. To var izmantot dažādu veidu problēmu risināšanai, sākot no vienkāršām programmām, konsoles aplikācijās līdz mākslīgajiem simulātoriem.

Django 4.0 [12] — tā ir programmatūra ar bagātīgām iespējām, kas ir piemērota sarežģītu vietņu un tīmekļa lietojumprogrammu izstrādei, kas rakstīta programmēšanas valodā *Python*.

pip [13] — pakešu pārvaldības sistēma, ko izmanto, lai instalētu un pārvaldītu programmatūrā Python rakstītas programmatūras pakotnes.

Nginx [14] — tā ir programmatūra ar atvērtu avota kodu, lai izveidotu vieglu un spēcīgu tīmekļa serveri. To izmanto arī kā pasta vai atpakaļejošu starpniekserveri.

Git [15] — izplatīta versiju kontroles sistēma, kas ļauj izstrādātājiem pārraudzīt failu izmaiņas un strādāt pie viena projekta kopā ar kolēģiem.

Github [16] — repozitoriju tiešsaistes hostinga serviss, kam ir visas paplašinātās versiju kontroles funkcijas un avota koda vadības funkcionalitāte, ir viss, kas atbalsta *Git* un pat vairāk.

Gunicorn [17] — pārvieto no *Nginx* saņemtos pieprasījumus formātā, kas var apstrādāt tīmekļa lietojumprogrammu, un nodrošina koda izpildi, ja nepieciešams.

MySQL [18] - kompānijas *Oracle* izstrādāta atvērtā koda relāciju datu bāzes pārvaldības sistēma;

Bootstrap [39] - *HTML* [52], *CSS* [53] un *JavaScript* [54] ietvars, kas nodrošina plašu klāstu ar dažādām lietotāja saskarnes komponentēm (pogas, modālie logi, navigācijas joslas un tml.), kā arī ērtu adaptīvā dizaina izstrādi. Nestrādā bez *jQuery*;

Visual Studio Code [21] — *Visual Studio Code* ir racionalizēts kodu redaktors ar tādu izstrādes darbību atbalstu kā atklādošana, uzdevumu izpilde un versiju kontrole. Tā mērķis ir nodrošināt tikai rīkus, kas izstrādātājiem ir nepieciešami ātrai koda izveides un atklādošanas ciklam, un padara sarežģītākas darba plūsmas pilnīgākas *IDEs*, piemēram, *Visual Studio IDE*.

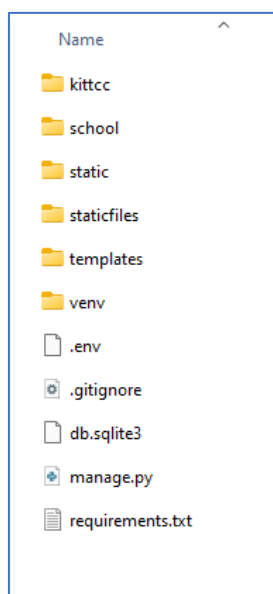
DB Management System [19] – programmu kopa, kas ļauj izveidot datu bāzi (DB) un manipulēt ar datiem (ievietot, atjaunināt, dzēst un atlasīt). Sistēma nodrošina datu glabāšanas drošību, uzticamību un integritāti, kā arī nodrošina rīkus datu bāzes administrēšanai. Dotajā tīmekļa lietotnē tika izmantota *PhpMyAdmin DB Managment System*.

Google Chrome [20] – *Google* izstrādāta pārlūkprogramma, kuras pamatā ir bezmaksas pārlūkprogramma *Chromium* [32] un *Blink* dzinējs [33], šajā projektā tika pielietota priekš, tīmekļa lietotnes vizuālām apskatam un funkcionalitātes pārbaudē.

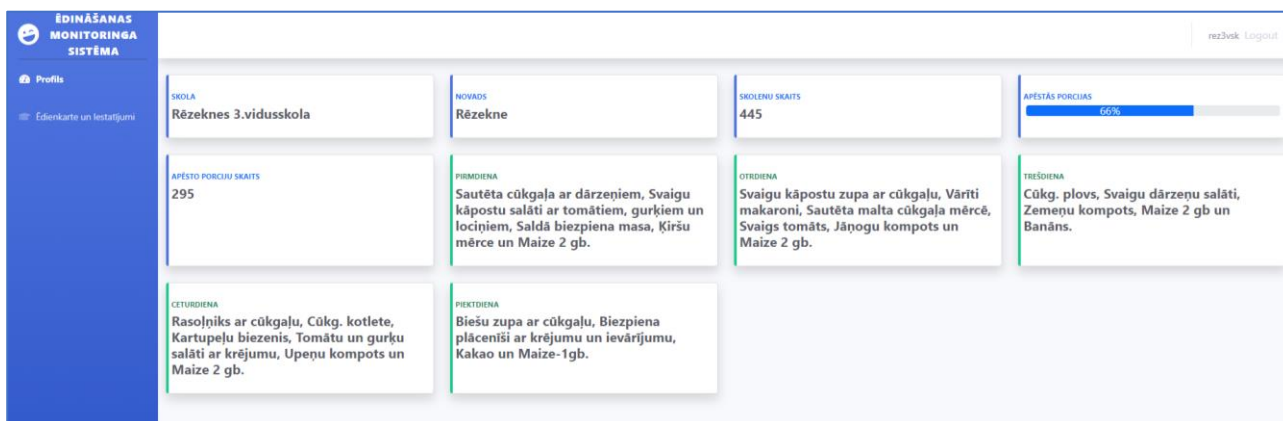
4. Izstrādes procesa apraksts

Bija izmantots gatavs ietvars *Django*, kas piedāvā gatavu mapju struktūru (skat. 4.1 attēlu).

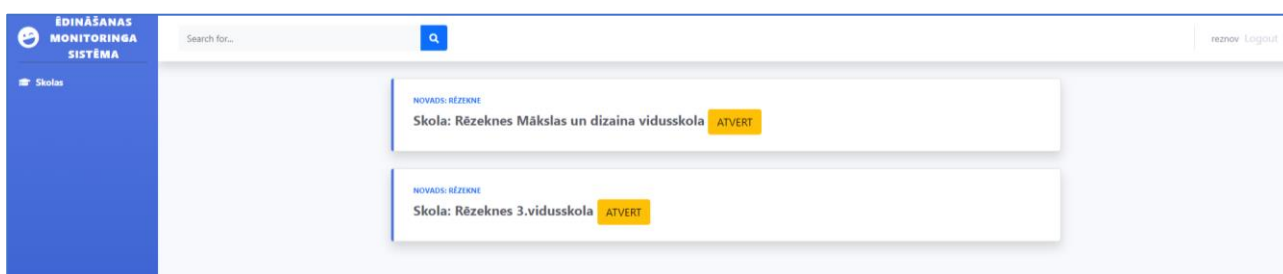
Tālāk notika klienta puses izstrāde (front-end). Tika pielietotas tehnoloģijas *Bootstrap*, *HTML* un *CSS*. Sākotnēji tika izveidota vizuālā tīmekļa lietotnes daļa, pēc kuras varēja orientēties kā tam ir jāizskatās, un kur būs funkcionālās pogas (skat. 4.2. att.).



4.1. attēls. Projekta resursu faili



4.2. attēls. Skolas lietotāja informācijas lapa



4.3. attēls. Pašvaldības lietotāja informācijas lapa

Sistēma ļauj izvēlēties no datubāzes konkrētas skolas, kurām sakrīt novads ar pašvaldību un uzspiežot pogu “Atvert” tiek atvērta informācija par konkrēto skolu, kā 4.3. attēlā un 4.5. attēlā – valdības profila piemērs.

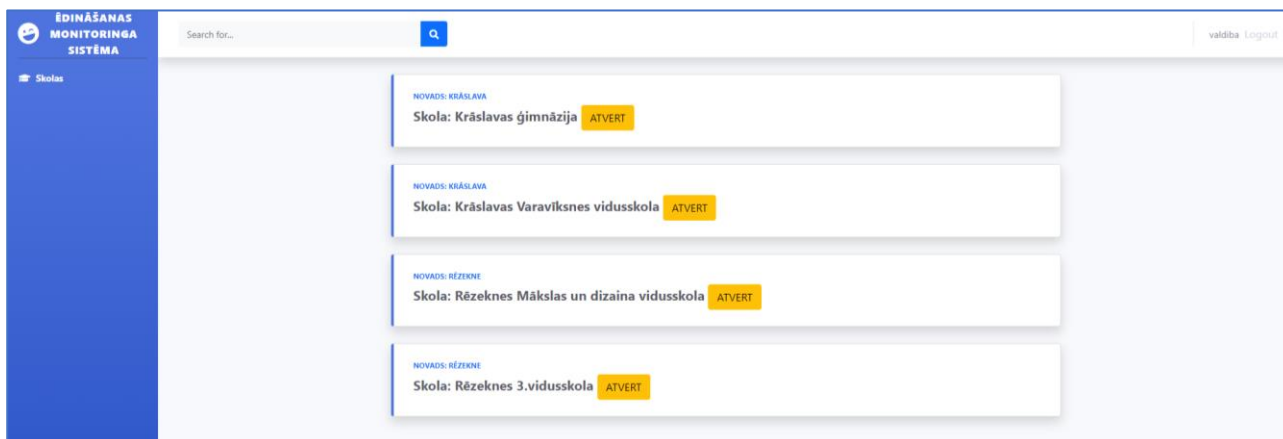
Lai organizētu datu ievadi un izvadi, tika izstrādāts datu modelis. *Django* ir iebūvēta datu bāzu migrācijas funkcionalitāte. Migrācijas faili tiek glabāti projektu direktoriijā un ļauj citiem sistēmas izstrādātājiem (ja tādi ir) mainīt datu bāzes struktūru, neizmantojot datu bāzu pārvaldības programmas (skat 4.4. att.), Datu bāzes tabulas User strukūras pirmskods (skat. 4.9 att.)

```

[Annas-MacBook-Pro:analytics_api annahuang$ docker-compose run dj python manage.py migrate auth
Starting rd ... done
Starting pges ... done
Operations to perform:
  Apply all migrations: auth
Running migrations:
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
[Annas-MacBook-Pro:analytics_api annahuang$ docker-compose run dj python manage.py migrate
Starting pges ... done
Starting rd ... done
Operations to perform:
  Apply all migrations: admin, api_app, auth, authtoken, contenttypes, django_celery_results, sessions
Running migrations:
  Applying api_app.0002_auto_20190817_2246... OK

```

4.4. attēls. Migrācijas procesa ilustrācija



4.5. attēls. *Valdības lietotājā informācijas lapa*

Šabloni ir tas, ko lietotājs redz. Django ietver jaudīgu veidņu dzinēju, kas ļauj izstrādātājam izmantot vairākas Python valodas konstrukcijas vienkāršās HTML veidnēs (ar `{{ }}` vai `{% %}`). Ir pieejami vairāki iebūvēti Django tagi (tags) un filtri (filters), kurus var arī viegli lietot. HTML veidnes paraugs — autorizācijas logs

```

1  {% if user.access == 2 %}
2  {% for el in students %}
3  {% if el.access == 0 %}
4
5
6  <div class="content">
7    <div class="card border-left-primary shadow h-100 py-2">
8      <div class="card-body">
9        <div class="row no-gutters align-items-center">
10         <div class="col mr-2">
11           <div class="text-xs font-weight-bold text-primary text-uppercase mb-1">
12             Novads: {{ el.novads }}</div>
13           <div class="h5 mb-0 font-weight-bold text-gray-800">Skola: {{ el.skola }}</div>
14           <a href="{% url 'school-detail' el.id %}" class="btn btn-warning">ATVERT</a>
15         </div>
16       </div>
17     <div class="col-auto">
18     </div>
19   </div>
20 </div>
21 </div>
22 </div>
23 </div>

```

4.6. attēls. *skolas.html: valdība, skolu izvade*

4.6. attēlā ir redzams fragments no faila “skolas.html”. Šis fails izvada informāciju par skolām, ja lietotāja pielaide ir vienāda ar divi, tādējādi tiek izvadīta informācija *HTML* lapā ar visām skolām, kas ir datubāzē. Nospiežot pogu *HTML* lapā (skat. 4.5. att.) “Atver”, notiek pārdresācija un noteiktās skolas id lapu kurā ir redzama visa statistika par skolu, skolas, skolēnu skaits, apēsto porciju skaits, skolas novadu un ēdienkarti uz vienu nedēļu. 4.7. attēlā ir redzams fragments no faila “skolas.html”. Šis fails izvada informāciju par skolām, ja lietotāja pielaide ir vienāda ar viens un skolas novads ir vienāds ar lietotāja novadu, tādējādi tiek izvadīta informācija *HTML* lapā, ja lietotāja pielaide nav vienāda ar viens vai divi, tad lietotāju pārnēs uz galveno lapu, jo skolām nav jābūt pielaides pie citu skolu datiem.

```

1  {% elif user.access == 1 %}
2  {% for el in students %}
3      {% if user.novads == el.novads %}
4          {% if el.access == 0 %}
5              <div class="content">
6                  <div class="card border-left-primary shadow h-100 py-2">
7                      <div class="card-body">
8                          <div class="row no-gutters align-items-center">
9                              <div class="col mr-2">
10                                 <div class="text-xs font-weight-bold text-primary text-uppercase mb-1">
11                                     Novads: {{ el.novads }}</div>
12                                 <div class="h5 mb-0 font-weight-bold text-gray-800">Skola: {{ el.skola }}
13                                 <a href="{% url 'school-detail' el.id %}" class="btn btn-warning">ATVERT</a>
14                                 </div>
15                             </div>
16
17                             <div class="col-auto">
18                                 </div>
19                             </div>
20                         </div>
21                     </div>
22                 </div>
23             {% endif %}
24         {% endif %}
25     {% endfor %}
26 {% else %}
27 <meta http-equiv="REFRESH" content="0;url=https://kittcc.net/accounts/profile">
28 {% endif %}
29 {% endblock %}

```

4.7 attēls skolas.html: novadi, skolu izvade

Datubāzes struktūra un dati tika pārbaudīti izmantojot *phpMyAdmin* [19] un pārlūk programmu “Google Chrome” [20]. Pie noklusētās datubāzes tabulas “*school_user*” tika pievienoti jauni lauki 4.8 attēla ir redzama daļa no datubāzes “*school_user*”, kurai tika pievienoti jauni lauki no *models.py*. Pēc jauno lauku pievienošanas ar *phpMyAdmin* palīdzību tika rediģēti jaunie lauki, kā arī to var darīt skolas tīmekļa lietotnē (skat. 4.13 att.)

skola	novads	skolenuskaits	apostasporcijas	pirmidiena	otrdiena	tresdiena	ceturtdiena	piekdiena	access
Krāslavas ģimnāzija	Krāslava	400	234	Svaigu kāpostu zupa ar cūkgaļu, Vārīti makaroni, S...	Rasolņiks ar cūkgaļu, Cūkg. kotiēte, Kartupeļu bie...	Biešu zupa ar cūkgaļu, Biezpiena plāceniši ar krēj...	Cūkg. plovš, Svaigu dārzeņu salāti, Zemeņu kompots...	Sautēta cūkgaļa ar dārzeņiem, Svaigu kāpostu salāt...	0
NULL	Krāslava			NULL	NULL	NULL	NULL	NULL	1
NULL	Rēzekne			NULL	NULL	NULL	NULL	NULL	1
Krāslavas Varavīksnes vidusskola	Krāslava	437	384	Rasolņiks ar cūkgaļu, Cūkg. kotiēte, Kartupeļu bie...	Cūkg. plovš, Svaigu dārzeņu salāti, Zemeņu kompots...	Biešu zupa ar cūkgaļu, Biezpiena plāceniši ar krēj...	Svaigu kāpostu zupa ar cūkgaļu, Vārīti makaroni, S...	Sautēta cūkgaļa ar dārzeņiem, Svaigu kāpostu salāt...	0
Rēzeknes Mākslas un dizaina vidusskola	Rēzekne	417	353	Sautēta cūkgaļa ar dārzeņiem, Svaigu kāpostu salāt...	Cūkg. plovš, Svaigu dārzeņu salāti, Zemeņu kompots...	Biešu zupa ar cūkgaļu, Biezpiena plāceniši ar krēj...	Rasolņiks ar cūkgaļu, Cūkg. kotiēte, Kartupeļu bie...	Svaigu kāpostu zupa ar cūkgaļu, Vārīti makaroni, S...	0
NULL	NULL			NULL	NULL	NULL	NULL	NULL	2
Rēzeknes 3.vidusskola	Rēzekne	445	295	Sautēta cūkgaļa ar dārzeņiem, Svaigu kāpostu salāt...	Svaigu kāpostu zupa ar cūkgaļu, Vārīti makaroni, S...	Cūkg. plovš, Svaigu dārzeņu salāti, Zemeņu kompots...	Rasolņiks ar cūkgaļu, Cūkg. kotiēte, Kartupeļu bie...	Biešu zupa ar cūkgaļu, Biezpiena plāceniši ar krēj...	0

4.8. attēls. Tabula *school_user* datubāzē

```

1 from django.db import models
2 from django.db import connections
3 from django.contrib.auth.models import AbstractUser
4 from .manager import UserManager
5
6 class User(AbstractUser):
7     email = None
8     skola = models.CharField(max_length=254)
9     novads = models.CharField(max_length=254)
10    skolenuskaitis = models.CharField(max_length=10)
11    apestasporcijas = models.CharField(max_length=10)
12    pirmdiena = models.TextField(null=True)
13    otrdiena = models.TextField(null=True)
14    tresdiena = models.TextField(null=True)
15    ceturdiena = models.TextField(null=True)
16    piekdiena = models.TextField(null=True)
17    access = models.SmallIntegerField(null=True)
18    objects = UserManager()
19    USERNAME_FIELD = 'username'
20    REQUIRED_FIELDS = []
21
22

```

4.9. attēls. *models.py* kods:

jauno lauku pievienošana pie noklusētās datubāzes “school_user”.

Vēl viena svarīga datne *Django* ietvarā ir “urls.py”, kur tiek aprakstīti “ceļi” (*URLs*). Šajā datnē ir aprakstīti visi iespējamie ceļi, kuri spēj lietotāju aizvest līdz galamērķim un atrādīt tam nepieciešamo informāciju ko pieprasīja lietotājs. *Django* ļauj izveidot vietrāžus URL, kā vien vēlaties, bez "ietvara" ierobežojumiem (skat. 4.10 att.)

```

1 urlpatterns = [
2     path('', auth_views.LoginView.as_view(template_name = "login.html"), name="login"),
3     path('accounts/logout', auth_views.LogoutView.as_view(), name="logout"),
4     path('about/', views.about, name="about"),
5     path('contact/', views.contact, name="contact"),
6     path('accounts/profile/', views.ProfileView.as_view(), name="profile"),
7     path('skolas/', views.home, name="home"),
8     path('skola/<int:pk>', views.SchoolDetailView.as_view(), name="school-detail"),
9     path('update/<int:pk>', views.SchoolUpdateView.as_view(), name="school-update"),
10 ] + static(settings.STATIC_URL, document_root=settings.STATIC_ROOT)

```

4.10. attēls. *urls.py* kods

Kontrolieris *views.py* pamatā izšķir divas pieejas koda rakstīšanai: funkciju kontrolleis (*def*) un klases kontrolleis (*class*). Atkarībā no sagaidāmā rezultāta izstrādātājs izlemj, kuru pieeju konkrētajā laikā izmantot.

def home(request) - dotā funkcija pāradresē lietotāju uz adresi *skolas.html*, ja, “ceļa” adrese ir vienāda ar norādīto *urls.py* (skat. 4.10 att.). Kā arī šajā funkcijā ir iebīvēta meklēšanas funkcija, ja skolas nosaukums, vai novada nosaukums sakrīt ar ievadīto ievades laukā, tad meklēšanas funkcija izdot arī visus laukus kuri sakrīt

Class SchoolUpdateView(UpdateView) – no datnes *models.py* (skat. 4.9 att.) tiek ņemti dati, kas ir ievadīti datubāzē un izvada tos uz ekrānu (skat 4.13 att.), tiek ņemts vērs, katra lietotāja pielāides līmeni, ja lietotājā pielāide ir 1 vai 2 tad, lietotājām nav pieejama poga, kas ļāutu pāriet pie Ēdienkartes un Iestatījumu redigēšanas.

```
1  {% if user.access == 0 %}
2      <li class="nav-item">
3          <a class="nav-link collapsed" href="{% url 'school-update' user.id %}">
4              <i class="fas fa-fw fa-graduation-cap"></i>
5              <span>Ēdienkarte un Iestatījumi</span>
6          </a>
7      {% endif %}
8
```

4.11 attēls **Lietotāju profila *HTML* lapas fragments**

Class SchoolUpdateSettingsView(UpdateView) – netiek pielietots dotajā tīmekļa lietotnē

Class SchoolDetailView(DetailView) – no datnes *models.py* (skat. 4.12 att.) tiek ņemti dati, kas ir ievadīti datubāzē un izvada tos uz ekrānu (skat 4.3 un 4.5 att.), tiek ņemts vērs, katra lietotāja pielāides līmeni, ja lietotājā pielāide ir 1 vai 2 tad, lietotājām ir pieejama poga, kas ļāutu pāriet pie Skolu monitorēšanas, ņemot vērs lietotājā pielāides līmeni, ja lietotājā pielāides līmenis ir vienāds ar 1, tad lietotājām izvadās uz ekrāna visas skolas, kas ir vienādā ar viņa novada nosaukumu (skat. 4.3 attēls), bet ja lietotājā pielāide ir vienāda ar 2 tad lietotājām tiek izvadītas visas skolas kas ir datubāzē izņemot, tos lietotājus kuriem pielāide ir vienādā ar 1 vai 2.

Python funkcijas, kas uzņem *http* pieprasījumus un atgriež *http* atbildi, piemēram, *HTML* dokumentus.

```

1 class SchoolUpdateView(UpdateView):
2     model = User
3     template_name = 'update_school_info.html'
4     context_object_name = 'schooluser'
5     form_class = UserForm
6
7 class SchoolUpdateSettingsView(UpdateView):
8     model = User
9     template_name = 'update_school_settings.html'
10    context_object_name = 'schooluser'
11    form_class = UserForm
12
13 class SchoolDetailView(DetailView):
14     model = User
15     template_name = 'details_view.html'
16     context_object_name = 'schooluser'
17
18 @login_required
19 def home(request):
20     search_query = request.GET.get('q','')
21
22     if search_query:
23         students = User.objects.filter(Q(skola__icontains=search_query) | Q(novads__icontains=search_query))
24     else:
25         students = User.objects.all()
26
27
28     return render(request,"skolas.html", {'students':students})
29
30
31
32 def login(request):
33     template = loader.get_template('login.html')
34     return HttpResponse(template.render({}, request))
35
36
37 def create(request):
38     error=''
39     if request.method == 'POST':
40         form = UserForm(request.POST)
41         if form.is_valid():
42             form.save()
43             return redirect('')
44         else:
45             error='Forma bija uzrakstīta kļūdaini'
46
47     form = UserForm()
48
49     data = {
50         'form':form,
51         'error':error
52     }
53     template = loader.get_template('update_school_info.html')
54     return HttpResponse(template.render({}, request),data)
55
56
57 @login_required
58 def about(request):
59     template = loader.get_template('about.html')
60     return HttpResponse(template.render({}, request))
61
62 @login_required
63 def contact(request):
64     template = loader.get_template('contact.html')
65     return HttpResponse(template.render({}, request))
66
67
68 class ProfileView(LoginRequiredMixin,TemplateView):
69     context_object_name = 'schooluser'
70     template_name = 'accounts/profile.html'

```

4.12. attēls. *views.py* kods

4.13. attēls. Ēdienkartes maiņa un iestatījumi

Šajā lāpā tiek dota iespēja ievadīt datus par ēdienkarti uz katru dienu, apēsto porciju skaitu un skolēnu skaitu, ko pēc tam var saglabāt un aplūkot skolas informācijas lapā (skat. 4.3. attēls).

```

1 class UserForm(ModelForm):
2     class Meta:
3         model = User
4         fields = ['pirmdiena', 'otrdiena', 'tresdiena', 'ceturdiena', 'piekdiena', 'apestasporcijas', 'skolenuskaitis']
5
6     widgets = {
7         "pirmdiena": TextInput(attrs={
8             'class': 'form-control',
9             'placeholder': 'Pirmdiena'
10        }),
11        "otrdiena": TextInput(attrs={
12            'class': 'form-control',
13            'placeholder': 'Otrdiena'
14        }),
15        "tresdiena": TextInput(attrs={
16            'class': 'form-control',
17            'placeholder': 'Trešdiena'
18        }),
19        "ceturdiena": TextInput(attrs={
20            'class': 'form-control',
21            'placeholder': 'Ceturtdiena'
22        }),
23        "piekdiena": TextInput(attrs={
24            'class': 'form-control',
25            'placeholder': 'Piektdiena'
26        }),
27        "apestasporcijas": NumberInput(attrs={
28            'class': 'form-control',
29            'placeholder': '0'
30        }),
31        "skolenuskaitis": NumberInput(attrs={
32            'class': 'form-control',
33            'placeholder': '0'
34        }),
35    }
36
37

```

4.14. attēls. forms.py kods

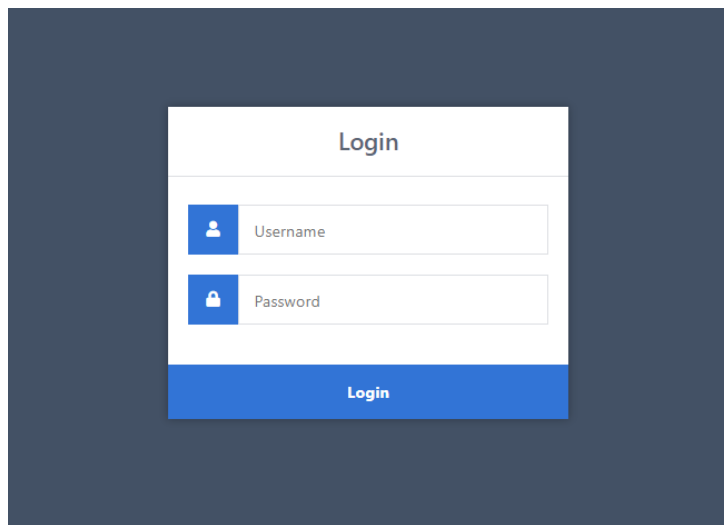
Tiek aprakstīta forma, kuru vēlamies izvadīt uz ekrānu, kura aizstāj mainīgos skolas ar parametriem no datubāzes, katrā no formām. (skat. 4.13. att.).


```

1  {% extends 'base.html' %}
2  {% load static %}
3
4  {% block title %}Login{% endblock %}
5
6  {% block content %}
7  {% load static %}
8  <!DOCTYPE html>
9  <html>
10     <head>
11         <meta charset="utf-8">
12         <meta http-equiv="X-UA-Compatible" content="IE=edge">
13         <title>Authentication</title>
14         <meta name="description" content="">
15         <meta name="viewport" content="width=device-width, initial-scale=1">
16         <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.7.1/css/all.css">
17         <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
18             rel="stylesheet" integrity="sha384-1BmE4kWbQ78iYhFldvKuhfTAU6auU8tT94WrHftjDbrCEXSU1oBoqyl2QvZ6jIW3"
19             crossorigin="anonymous">
20         <link rel="stylesheet" href="{% static 'style.css' %}">
21     </head>
22     <body>
23
24         <div class="login">
25             <h1>Login</h1>
26             <form method="POST">
27                 {% csrf_token %}
28
29                 <label for="username">
30                     <i class="fas fa-user"></i>
31                 </label>
32                 <input type="text" name="username" placeholder="Username" id="username" required>
33                 <label for="password">
34                     <i class="fas fa-lock"></i>
35                 </label>
36                 <input type="password" name="password" placeholder="Password" id="password" required>
37                 <input type="submit" value="Login">
38             </form>
39         </div>
40     </body>
41 </html>
42
43 {% endblock %}

```

4.15 attēls. Autorizācijas *HTML* lapa

A login form UI mockup centered on a dark blue background. The form is a white rectangle with a blue header bar at the top containing the word "Login" in white. Below the header, there are two input fields. The first field has a blue icon of a person on the left and the text "Username" inside. The second field has a blue icon of a padlock on the left and the text "Password" inside. At the bottom of the form is a solid blue button with the word "Login" in white text.

Login

Username

Password









Login

4.16. attēls **Ielogošanās logs lietotājām**

5. Instalācijas izstrāde

5.1 Virtuālas vides instalācija

Pirms sākt sistēmas uzstādīšanu ir jāizveido virtuāla vide. Pirmais solis izveidot jaunu direktoriju, kurā tiks glabāts *Django* projekts. Tālāk, jāinstalē *Python* – kad izstrāde tika uzsākta, pēdējā versija bija *Python 3.9.10* (<https://www.python.org/downloads/>) – tas ir arī jālejupielādē un jāinstalē (skat. 5.1. att.).

Release version	Release date		Click for more
Python 3.9.13	May 17, 2022	 Download	Release Notes
Python 3.10.4	March 24, 2022	 Download	Release Notes
Python 3.9.12	March 23, 2022	 Download	Release Notes
Python 3.10.3	March 16, 2022	 Download	Release Notes
Python 3.9.11	March 16, 2022	 Download	Release Notes
Python 3.8.13	March 16, 2022	 Download	Release Notes
Python 3.7.13	March 16, 2022	 Download	Release Notes
Python 3.9.10	Jan. 14, 2022	 Download	Release Notes
View older releases			

5.1.attēls. Python lejupielādes logs

Kad esat pārliecinājies, ka *Python* ir veiksmīgi instalēts, var atvērt komandrindu un ievadīt tajā komandu *python -V* (skat. 5.2. att.).

```
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hara1>python -V
Python 3.9.13

C:\Users\hara1>_
```

5.2.attēls. Pārbaude, ka Python ir sekmīgi uzinstalēts

Tālāk, caur to pašu komandrindu jāiziet līdz jaunizveidotai direktorijai, izmantojot komandu `cd` (skat. 5.3. att.).

```
Microsoft Windows [Version 10.0.22000.675]
(c) Microsoft Corporation. All rights reserved.

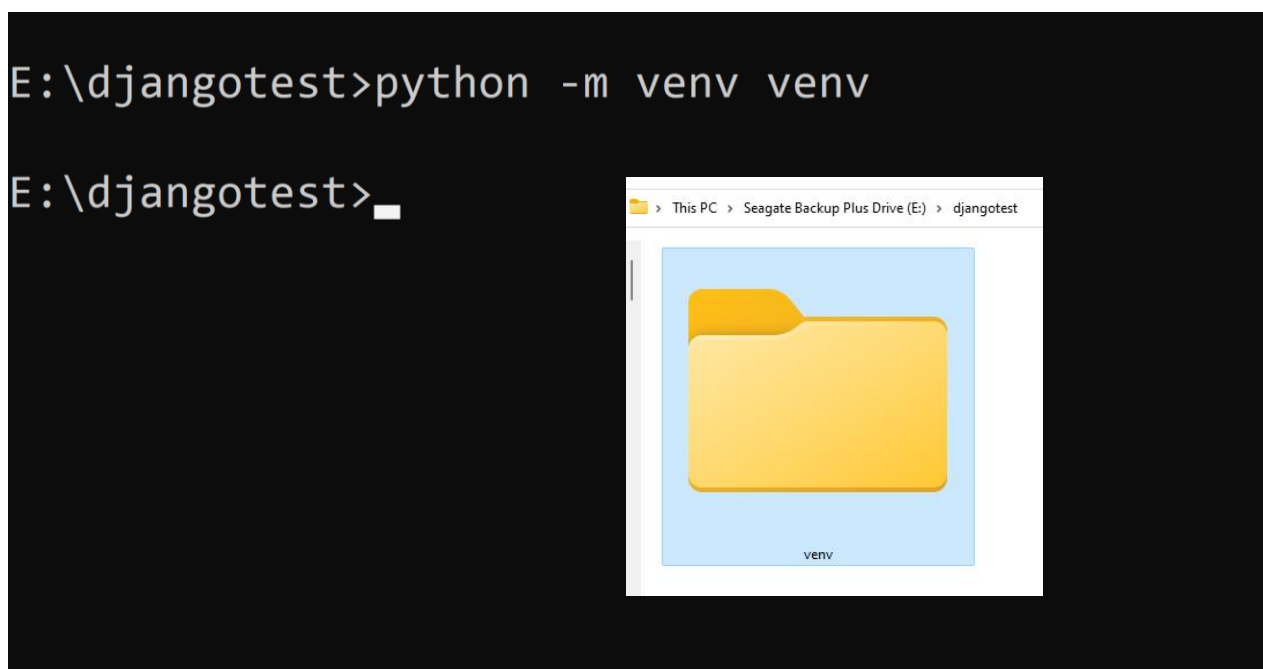
C:\Users\haral>e:

E:\>cd django test

E:\django test>_
```

5.3.attēls. Pārvietošanās pie direktorijas, kur tiks izveidota virtuālā vide

Kad pie nepieciešamās direktorijas ir nokļūts, jāizmanto *Python* iestrādāto komandu *venv* (*virtual environment*) šādā vārdu kombinācijā: `python -m venv venv` (skat. 5.4. att.) – šajā gadījumā, šī komanda direktorijā *django test* izveidos direktoriju ar nosaukumu *venv*.

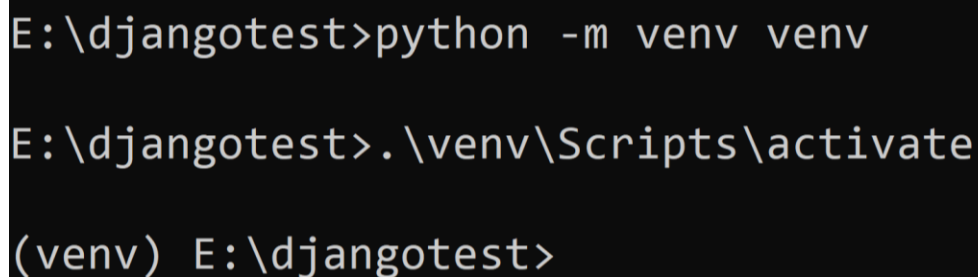


5.4.attēls. Virtuālās vides izveidošanas ilustrācija

Lai pārliecināties, ka virtuālā vide tika izveidota korekti, to vajag aktivizēt – tādā veidā pārbaudot tās darbību. Tas darāms ar komandu: `.\venv\Scripts\activate` (skat. 5.5 att.).

Ja tā tiks aktivizēta, sagaidāms, ka pirms lokāla diska burta (*E:\djangotest*) parādīsies iekavās uzraksts

(venv) – tas nozīmēs, ka virtuāla vide ir aktivizēta un darbojas korekti.



```
E:\djangotest>python -m venv venv  
E:\djangotest>.\venv\Scripts\activate  
(venv) E:\djangotest>
```

5.5.attēls. Virtuālās vides aktivizācijas ilustrācija

Visas tālākas darbības jau notiek *Visual Studio Code* piedāvātajā termināla logā, kas ir analogs *Windows* komandrindai.

5.2 Django ietvara instalācija

Kā jau tika minēts iepriekšējās nodaļas beigās, tālākais darbs notiks *Visual Studio Code* izstrādes vidē un tajā pieejamajā termināla logā. Iepriekšējā nodaļā tika uzinstalēta virtuālā vide, kuru tagad jāpapildina ar *Django* ietvaru, kas arī skaitās par pakotni. Kad *Visual Studio Code* ir atvērts un tajā atvērta direktorijs, *Visual Studio Code* automātiski aktivizēs iepriekšējā solī izveidoto virtuālo vidi. Viss, ko atliek izdarīt izstrādes sākumam, ir komanda: *pip install Django* (skat. 5.6. att.).

```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE

(venv) E:\djangotest>pip install Django
Collecting Django
  Downloading Django-4.0.5-py3-none-any.whl (8.0 MB)
    8.0/8.0 MB 17.7 MB/s eta 0:00:00
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.2-py3-none-any.whl (42 kB)
Collecting tzdata
  Using cached tzdata-2022.1-py2.py3-none-any.whl (339 kB)
Collecting asgiref<4,>=3.4.1
  Downloading asgiref-3.5.2-py3-none-any.whl (22 kB)
Installing collected packages: tzdata, sqlparse, asgiref, Django
Successfully installed Django-4.0.5 asgiref-3.5.2 sqlparse-0.4.2 tzdata-2022.1
WARNING: You are using pip version 22.0.4; however, version 22.1.2 is available.
You should consider upgrading via the 'E:\djangotest\venv\Scripts\python.exe -m pip install --upgrade pip' command.

(venv) E:\djangotest>
```

5.6.attēls. *Django* ietvara instalācijas ilustrācija

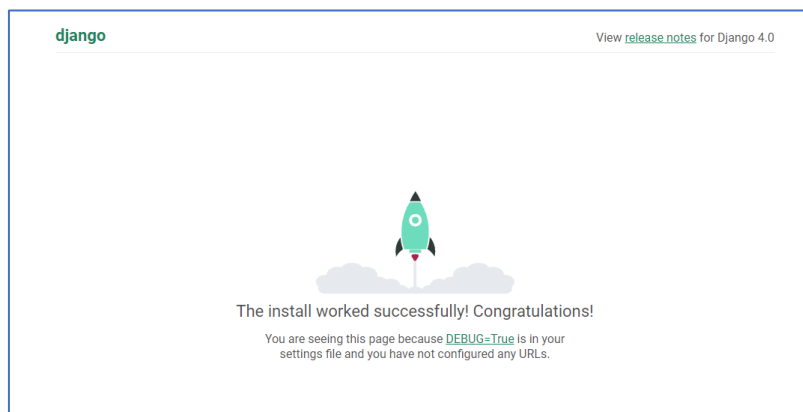
Django pašlaik ir tukšs. Lai to mainīt un sākt izstrādi, termināla logā jāievada komanda:°

django admin startproject skola, kur pēdējais vārds ir kopējās direktorijās nosaukums (skat. 5.6.att.). Blakus direktorijai *venv* parādīsies vēl viena – *skola*, kura satur balsta konfigurāciju topošajai.

```
(venv) E:\djangotest>django-admin startproject skola
```

5.7.attēls. *Django* papildināšana ar jaunu direktoriju

Ja tagad ievadīt komandu *python manage.py runserver*, palaidīsies virtuālais lokālais serveris ar sākuma skatu, kad programmas kods vēl netika kaut kādā veidā mainīts (skat. 5.8. att.).



5.8.attēls. *Django* lokālā servera sekmīga palaišana

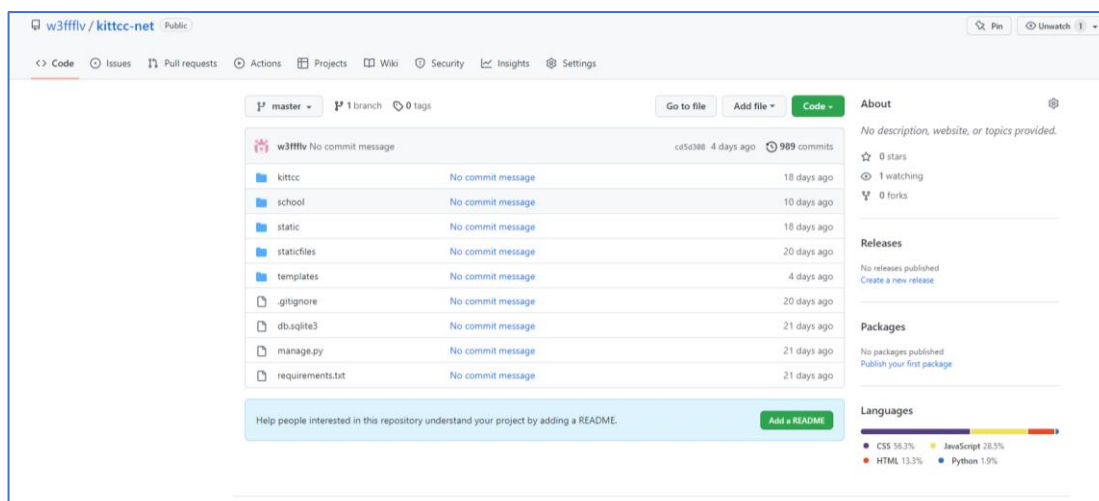
Pēdējais solis, lai sākt pilnvērtīgu izstrādi ir kopējai konfigurācijas direktorijai izveidot to, kurā tad arī notiks lielākums mijiedarbības ar programmas kodu. Tam ir paredzēta vēl viena komanda: *python manage.py startapp school*, kur pēdējais vārds definē ,kā sauksies direktorija ar autora izstrādājamo lietotni (skat. 5.9 att.).

```
(venv) E:\djangotest\skola>django-admin startapp school
```

5.9 attēls *Django* lietotnes pirmā komanda

5.2 Projekta koda augšupielāde

Izstrādājamās sistēmas koda glabāšanai tika izmantots *GitHub* serviss. *GitHub* ir *Git* veidīgs repozitoriju pārvaldības risinājums, kas dod iespēju centralizēti pārvaldīt repozitorijus un kontrolēt projekta izstrādes plūsmu. Projekts tika augšupielādēts uz *Github* [28] servisu, kur repozitoriju var lejupielādēt visi.



5.10. attēls Projekta repozitorijs Github tīmekļa lietotnē

5.3 Projekta pārņemšana

Pēc projekta augšupielādes no repozitorija, projektu var pārņemt projekta mapē, kas ir attēlota 5.4. attēla. Sākotnēji izdzēšot visus failus, no mapes, un pārnesot šī projekts saturu, kas ir attēlots 5.10. attēlā.

5.4 Darbs ar datu bāzi

Izstrādājot projektu darbam ar Mysql datu bāzi, tika izmantots PhpMyAdmin rīks [19]. Tas ir bezmaksas rīks, kas darbojas kā tīmekļa lietojumprogramma un nodrošina plašas PhpMyAdmin datu bāzes pārvaldības iespējas:

- PhpMyAdmin servera konfigurēšana
- lietotāju un viņu privilēģiju pārvaldība
- datubāzes, tabulu, kolonnu un indeksu izveide, rediģēšana un dzēšana

Secinājumi un priekšlikumi

Secinājumi:

1. Izveidojot vienu *HTML* šablonu, to var izmantot vairākās lapās.
2. Lietotnes izstrāde ir relatīvi viegla, izmantojot *Django* ietvaru.
3. Pēc rediģēšanas ir nepieciešams restartēt serveri, ja faila formāts bija *.py*, lai rediģējamais fails strādātu korekti.
4. Visi uzdevumi bija izpildīti, darba mērķis ir sasniegts.

Priekšlikumi:

Ieviest sistēma uzlabojumus, lai varētu glabāt datus par apēstām porcijām, apēsto porciju skaitu un ēdienkarti par nedēļu.

Kopsavilkums

Šajā kvalifikācijas darbā tika aprakstīts ‘Skolu ēdināšanas monitoringa sistēma’ tīmekļa lietojumprogrammu. Darbs sastāv no 5 sadaļām un 7 apakšsadaļām. Katrā sadaļā tiek apskatīta noteikta projekta tēma, no kodēšanas procesa apraksta līdz lietotāja instrukcijas izstrādei un tā tālāk. Kvalifikācijas darba ievadā tika noteikti 7 mērķi darba pilnveidošanai, kuri tika izpildīti – analizēt esošu situāciju, paskaidrot izstrādes tehnoloģijas izvēli, aprakstīt izstrādes procesu, izstrādāt lietotāja instrukciju, izstrādāt lietotnes instalāciju un veidot secinājumus un priekšlikumus.

Šajā darbā aprakstīta tīmekļa lietojumprogrammu ir saistīta ar statistikas vākšanas un monitoringa. Izstrādājamā tīmekļa lietojumprogrammu ļauj lietotājiem apskatīt statistiku par skolu un rediģēt to. Šī tīmekļa lietotne var interesēt lietotājus, kas ir saistīti ar skolu ēdināšanas un monitoringa sfēru.

Summary

This qualification work described the “School Nutrition Monitoring System” web application. Work consists of 5 sections and 7 subsections. Each section looks at a specific project topic, from the description of the coding process to the development of the user instruction, and so on. The Qualification Work Introduction identified 7 objectives for job development that were met: analysing an existing situation, explaining the choice of development technology, describing the development process, developing user instruction, developing an application installation and creating conclusions and proposals.

This work describes a web application related to the collection and monitoring of statistics. The web application to be developed allows users to view and edit school statistics. This web app may interest users associated with the school catering and monitoring sphere.

Literatūra

1. <https://github.com/alexchevsky/smysl-io>
2. https://www.obeythetestinggoat.com/book/appendix_III_provisioning_with_ansible.html
3. https://www.obeythetestinggoat.com/book/chapter_making_deployment_production_ready.html
4. https://www.youtube.com/watch?v=6K83dgjkQNw&ab_channel=%D0%93%D0%BE%D1%88%D0%B0%D0%94%D1%83%D0%B4%D0%B0%D1%80%D1%8C
5. https://docs.djangoproject.com/en/4.0/R8qH5hd4UAPlt5r1BrsdNHQuaGhg2Zd6J43CkaAn1qEALw_wcB
6. <https://realpython.com/django-setup/>
7. <https://www.tangowithdjango.com/>
8. <https://djangobook.com/>
9. https://www.youtube.com/watch?v=8QnMxrNT0-8&t=732s&ab_channel=%D0%A8%D0%BA%D0%BE%D0%BB%D0%B0itProger%2F%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5
10. https://www.youtube.com/watch?v=G3TLZH9bMUE&ab_channel=%D0%A8%D0%BA%D0%BE%D0%BB%D0%B0itProger%2F%D0%9F%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B5
11. <https://www.python.org/>
12. <https://www.djangoproject.com/>
13. <https://pypi.org/project/pip/>
14. <https://www.nginx.com/>
15. <https://git-scm.com/>
16. <https://github.com/>
17. <https://gunicorn.org/>
18. <https://www.mysql.com/>
19. <https://www.phpmyadmin.net/>
20. https://www.google.com/chrome/?brand=YTUH&gclid=Cj0KCQjwqPGUBhDwARIsANNwjV5C4gieJlp1rX_1bu-gJX2vUV6YhR82vbFU1hUlh0zKDLLMp28E53gaAucmEALw_wcB&gclid=aw.ds

21. <https://code.visualstudio.com/>
22. <https://info.winnowsolutions.com/food-waste-management>
23. https://www.youtube.com/watch?v=dBXoKXuxjyE&ab_channel=WinnowSolutions
24. <https://www.marketman.com/restaurant-waste-software/>
25. https://www.youtube.com/watch?v=uFwPz8GwegM&ab_channel=MarketMan
26. <https://www.jamix.com/food-waste-tracking/>
27. <https://www.youtube.com/channel/UCh4ZeHpJE2qAIA-plpfELdA>
28. <https://github.com/w3fflv/kittcc-net>
29. <https://www.digitalocean.com/>
30. <https://www.hostwinds.com/>
31. <https://hromadske.ua/ru/posts/lyudi-za-god-vybrasyvayut-931-mln-tonn-edy-eto-ravnyaetsya-vesu-23-mln-gruzovikov-kotorymi-mozhno-sem-raz-obognut-zemlyu>
32. <https://www.chromium.org/chromium-projects/>
33. <https://www.chromium.org/blink/>
34. <https://ru.hostings.info/schools/bazy-dannyh.html>
35. <https://help.reg.ru/hc/ru/articles/4408047763217-%D0%A7%D1%82%D0%BE-%D1%82%D0%B0%D0%BA%D0%BE%D0%B5-MySQL>
36. <https://www.appdynamics.com/topics/database-management-systems>
37. <https://itglobal.com/ru-ru/company/glossary/subd-sistema-upravleniya-bazami-dannyh/>
38. <https://getbootstrap.com/>
39. <https://icons.getbootstrap.com/>
40. <https://fontawesome.com/>
41. <https://itchief.ru/bootstrap/introduction>
42. <https://freehost.com.ua/faq/wiki/chtotakoe-phpmyadmin/>
43. <https://www.atlassian.com/ru/git/tutorials/what-is-git>
44. <https://www.hostinger.com.ua/rukovodstva/chtotakoe-github/>
45. <https://ubuntu.com/>
46. <https://itchief.ru/bootstrap/introduction>
47. <https://ubuntu.com/>
48. <https://www.techtarget.com/searchdatamanagement/definition/database-management-systemhttps://ubuntu.com/>
49. <https://rus.err.ee/1608132295/ezhegodno-vo-vsem-mire-vybrasyyvaetsja-bolee-900-mln-tonn-produktov-pitanijahttps://ubuntu.com/>

- 50. <https://rus.err.ee/1608132295/ezhegodno-vo-vsem-mire-vybrasyvaetsja-bolee-900-mln-tonn-produktov-pitanija>
- 51. <https://www.unep.org/thinkeatsave/get-informed/worldwide-food-waste>
- 52. <https://developer.mozilla.org/en-US/docs/Web/HTML>
- 53. <https://developer.mozilla.org/en-US/docs/Web/CSS>
- 54. <https://www.javascript.com/>

Dokumentārā lapa

Kvalifikācijas darbs “**Skolu ēdināšanas monitoringa sistēma**” ir izstrādāts RTA inženieru fakultātē, pirmā līmeņa profesionālās augstākās izglītības studiju programmā “Programmēšana un datortīklu administrēšana”.

Ar savu parakstu apliecinu, ka darbs veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Darba autors: Haralds Glaudāns

_____._____._____.

(datums)

_____ (paraksts)

Darba vadītājs: Dr.sc.ing. Sergejs Kodors

Recenzents:

Darbs aizstāvēts kvalifikācijas darba gala (valsts) pārbaudījuma komisijas sēdē
prot. Nr. _____

Datums: _____._____._____.

Novērtējums _____

Komisijas sekretārs: _____
(vārds uzvārds, paraksts)