|  |  |  |
| --- | --- | --- |
| viko_s_k | | |
| VILNIAUS KOLEGIJA  ELEKTRONIKOS IR INFORMATIKOS FAKULTETAS  PROGRAMINĖS ĮRANGOS KATEDRA | | |
|  |  |  |
| **PROGRAMŲ SISTEMŲ INŽINERIJA** | | |
| PROJEKTAS  6531BX028 PI20B | | |
| STUDENTAS |  | ROMUALDAS GIRČYS |
|  |  |  |
| DĖSTYTOJA |  | JOANA KATINA |
|  |  |  |
|  |  |  |
|  |  |  |
| 2022 | | |

**TURINYS**

[ĮVADAS 4](#_Toc124134852)

[1. UŽDUOTIES ANALIZĖ IR DARBO PLANAS 5](#_Toc124134853)

[1.1. Sistemos vizija 5](#_Toc124134854)

[1.2. Pradiniai programų sistemos reikalavimai 6](#_Toc124134855)

[1.3. Projekto vykdytojai 7](#_Toc124134856)

[1.4. Projekto planas 8](#_Toc124134857)

[1.5. Projekto rizikų analizė 9](#_Toc124134858)

[2. NAUDOTOJO REIKALAVIMAI 10](#_Toc124134859)

[3. SISTEMOS REIKALAVIMAI 11](#_Toc124134860)

[3.1. FUNKCINIAI REIKALAVIMAI 11](#_Toc124134861)

[3.2. NEFUNKCINIAI REIKALAVIMAI 15](#_Toc124134862)

[4. ARCHITEKTŪROS APRAŠYMAS 16](#_Toc124134863)

[5. PROGRAMŲ SISTEMOS PROJEKTINĖ DALIS 17](#_Toc124134864)

[5.1. FUNKCINIŲ REKALAVIMŲ TRASAVIMO LENTELĖ 17](#_Toc124134865)

[5.2. DUOMENŲ BAZĖS MODELIS 18](#_Toc124134866)

[5.3. KLASIŲ DIAGRAMA / ANALIZĖS OBJEKTO DIAGRAMA 19](#_Toc124134867)

[5.4. SISTEMOS ELGSENOS DIAGRAMOS 20](#_Toc124134868)

[6. REALIZAVIMO ĮRANKIŲ IR PRIEMONIŲ APRAŠYMAS 25](#_Toc124134869)

[7. TESTAVIMO PLANAS 26](#_Toc124134870)

[8. NAUDOTOJO INSTRUKCIJA 27](#_Toc124134871)

[IŠVADOS 28](#_Toc124134872)

**PAVEIKSLŲ SĄRAŠAS**

[1 pav. Svečias naudotojas 5](#_Toc124136058)

[2 pav. Registruotas naudotojas 5](#_Toc124136059)

[3 pav. Administratorius 5](#_Toc124136060)

[4 pav. Panaudos atvejų diagrama 6](#_Toc124136061)

[5 pav. Darbų išskaidymo struktūra 8](#_Toc124136062)

[6 pav. Ganto diagrama 8](#_Toc124136063)

[7 pav. Panaudos atvejų diagrama 11](#_Toc124136064)

[8 pav. MVC architektūra 16](#_Toc124136065)

[9 pav. Esybių ryšių diagramą 18](#_Toc124136066)

[10 pav. Klasių diagrama 19](#_Toc124136067)

[11 pav. Veiklos diagrama prisijungimui 20](#_Toc124136068)

[12 pav. Veiklos diagrama registracijai 20](#_Toc124136069)

[13 pav. Veiklos diagrama blogo peržiūrai 21](#_Toc124136070)

[14 pav. Veiklos diagrama keisti informaciją 21](#_Toc124136071)

[15 pav. Veiklos diagrama keisti informaciją 22](#_Toc124136072)

[16 pav. Veiklos diagrama sukurti blogus 22](#_Toc124136073)

[17 pav. Veiklos diagrama keisti blogus 23](#_Toc124136074)

[18 pav. Veiklos diagrama pašalinti naudotoją 23](#_Toc124136075)

[19 pav. Veiklos diagrama pašalinti blogą 24](#_Toc124136076)

[20 pav. Failas „requirements.txt “ 25](#_Toc124136077)

[21 pav. Programos aplankalo atidarymas 27](#_Toc124136078)

[23 pav. Terminalo paleidimas 27](#_Toc124136079)

[24 pav. Terminalo komanda 28](#_Toc124136080)

[25 pav. Nuoroda į tinklalapį 28](#_Toc124136081)

[26 pav. Svetainė naršyklėje 29](#_Toc124136082)

ĮVADAS

Darbo tikslas sukurti blogų svetainę naudotojams peržiūrėti ir skelbti savo blogus. Darbui bus naudojama „Pythone“ programavimo kalba ir „Djnago“ karkasas.

**Darbo tikslas –** sukurti svetaine naudojant „Pythono“ programavimo kalbą ir „Django“ karkasą

**Darbo uždaviniai:**

1. Analizė
2. Projektavimas
3. Įgyvendinimas
4. Testavimas
5. UŽDUOTIES ANALIZĖ IR DARBO PLANAS
   1. Sistemos vizija

Kuriamos sistemos paskirti leisti naudotojams peržiūrėti kitų naudotojų sukurtus blogus, patiems juos kurti ir paskelbti tinklalapyje. Sistema bus kuriama naudojant MVC (ang. Model View Controller) architektūrą.

A picture containing shoji, tiled

Description automatically generated

pav. Svečias naudotojas

Pirmas naudotojas sistemoje yra svečias naudotojas galintis tik apsilankyti tinklalapyje ir peržiūrėti blogus.

A picture containing text, shoji, public, tiled

Description automatically generated

pav. Registruotas naudotojas

Antras sistemos naudotojas yra registruotas naudotojas galinti apsilankyti tinklalapyje ir kurti naujus blogus.

A picture containing text, shoji, tiled

Description automatically generated

pav. Administratorius

Paskutinis naudotojas yra administratorius tvarkanti tinklalapio naudotojus ir blogus

* 1. Pradiniai programų sistemos reikalavimai

Sistemos pagrindine reikalavimai sudarys: prisijungimas, užsiregistravimas, blogų peržiūra, naudotojo informacijos keitimas, blogų sukūrimas, keitimas, naudotojų pašalinimas ir blogų pašalinimas.

Diagram

Description automatically generated

pav. Panaudos atvejų diagrama

Svečias galės prisijungti prie sistemos, užsiregistruoti ir peržiūrėti blogus. Registruotas naudotojas galės peržiūrėti blogus keisti savo informaciją, sukurti blogą ir keisti blogą. Administratorius galės pašalinti naudotojus ir pašalinti blogus.

* 1. Projekto vykdytojai

Projektas bus atliekamas individuali vieno studento.

|  |  |
| --- | --- |
| **STUDENTAS** | **PAREIGOS** |
| Romualdas Girčys | Projekto vadovas.  Programuotojas. |

Toliau pateikta pareigos ir jų atsakomybės.

|  |  |
| --- | --- |
| **PAREIGOS** | **ATSAKOMYBĖS** |
| Projekto vadovas | Vadovauja ir prižiūri, kad projektas būti sklandus. |
| Programuotojas | Programuoja paskirtus darbus. |

Komunikavimo plano nereikės nes projektas atliekamas individualiai.

* 1. Projekto planas

Pasirinktas krioklio programų kūrimo procesas nes kuriamas projektas yra nedidelis ir nesudėtingas.

Diagram

Description automatically generated

pav. Darbų išskaidymo struktūra

Sudaryta darbų išskaidymų struktūra rodanti pagrindines užduotis 1. planavimą, 2. kodo rašymą ir 3. testavimą. Mažesnės užduotis yra: 1.1 programos kūrimo procesas pasirinktas krioklio, 1.2 programos architektūra yra MVC, 2.1 Paskiros naudotojai, 2.2 Blogai ir jų informacija, 2.3 Naudotojai ir jų asmeninė informacija, 3.1 Unit testai sudaryti testuoti sistemą, 3.2 Sistemos testavimas iš naudotojo pusės.

A picture containing timeline

Description automatically generated

pav. Ganto diagrama

Ganto diagramoje parodytos užduotys sistemoje kurias reikia atlikti. „App mysite“ pagrindinė programa sujungianti viska, „App blog“ sauganti informacija apie blogus, „App personal“ saugo naudotojų asmeninę informaciją, „App account“ naudotojų informaciją sujungia su paskirą, „Templates“ saugo pagrindinius šablonus, „Media files“ saugo naudotojų ir blogų nuotraukas, „Unit test“ testai sistemai, „User acceptence testing“ sistemos testavimas iš naudotojo pusės.

* 1. Projekto rizikų analizė

Projekto sudarytos rizikos ir jų sumažinimo metodai.

|  |  |
| --- | --- |
| **Rizika** | **Rizikos valdymo strategija** |
| Reikalavimų pasikeitimas programavimo metu | Pasikeitimų realizacijos atidėjimas |
| Nerealistinis laiko ir sąnaudų įvertinimas | Praeityje vykdytų projektų analizė |
| Nepatogios naudotojo sąsajos keitimas | Detali naudotojo atliekamų užduočių analizė |

1. NAUDOTOJO REIKALAVIMAI

Naudotojų sistemoje yra 3 aktoriai: svečiai, registruoti naudotojai, administratorius

1. Svečias turi turėti galimybę prisijungti prie sistemos.
2. Svečias turi turėti galimybe užsiregistruoti prie sistemos.
3. Svečias ir naudotojas turi turėti galimybe peržiūrėti blogus.
4. Registruotas naudotojas turi turėti galimybe sukurti blogą.
5. Registruotas naudotojas turi turėti galimybe keisti blogą.
6. Registruotas naudotojas turi turėti galimybe keisti asmenine informaciją
7. Administratorius turi turėti galimybe pašalinti naudotoją.
8. Administratorius turi turėti galimybe pašalinti blogą.
9. SISTEMOS REIKALAVIMAI
   1. FUNKCINIAI REIKALAVIMAI

Sistemos pagrindiniai reikalavimai sudarys: prisijungimas, užsiregistravimas, blogų peržiūra, naudotojo informacijos keitimas, blogų sukūrimas, keitimas, naudotojų pašalinimas ir blogų pašalinimas.

Diagram

Description automatically generated

pav. Panaudos atvejų diagrama

Svečias galės prisijungti prie sistemos, užsiregistruoti ir peržiūrėti blogus. Registruotas naudotojas galės peržiūrėti blogus keisti savo informaciją, sukurti blogą ir keisti blogą. Administratorius galės pašalinti naudotojus ir pašalinti blogus.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **FUNKCINIS REIKALAVIMAS** | **NAUDOTOJO REIKALAVIMAS** | **PANAUDOS ATVEJIS** | **PRIORITETAS** |
| FR1 | Sistema turi leisti svečiams prisijungti prie sistemos, naudojant elektronini paštą ir slaptažodį | 1 | UC01 | Būtinas |
| FR2 | Sistema turi leisti svečiams prisiregistruoti prie sistemos, naudojant elektronini paštą, vardas ir slaptažodį | 2 | UC02 | Būtinas |
| FR3 | Sistema turi leisti svečiams ir registruotoms naudotojams peržiūrėti blogus | 3 | UC03 | Būtinas |
| FR4 | Sistema turi leisti registruotoms naudotojams keisti asmeninę informaciją | 4 | UC04 | Būtinas |
| FR5 | Sistema turi leisti registruotiems naudotojams sukurti blogus | 5 | UC05 | Būtinas |
| FR6 | Sistema turi leisti registruotiems naudotojams keisti blogus | 6 | UC06 | Būtinas |
| FR7 | Sistema turi leisti administratoriui pašalinti naudotojus | 7 | UC07 | Būtinas |
| FR8 | Sistema turi leisti administratoriui pašalinti blogus | 8 | UC08 | Būtinas |

|  |  |
| --- | --- |
| **Pavadinimas** | Prisijungti sistemoje |
| **ID** | UC01 |
| **Trumpas aprašymas** | Svečias gali prisijungti sistemoje |
| **Naudotojas (-ai)** | Svečias |
| **Prieš-sąlygos** | Veikia internetinis ryšys |
| **Pagrindinis scenarijus** | 1. Svečias inicijuoja prisijungimą 2. Sistema pateikia dialogo langą duomenų įvedimui. 3. Svečias įveda informaciją ir patvirtina įvedimą 4. Sistema patvirtina informaciją. |
| **Alternatyvūs scenarijai** | 4.a Neteisinga informacija.  4.a.1 Sistema informuoja naudotoją.  4.a.2 Naudotojas pakeičia informaciją.  4.a.3 Sistema patvirtina informaciją |
| **Po-sąlygos** | Svečias prisijungia prie sistemos |
| **Papildoma informacija** | Informacija apie klientą:  elektrinis paštas, slaptažodis |

|  |  |
| --- | --- |
| **Pavadinimas** | Registruotis sistemoje |
| **ID** | UC02 |
| **Trumpas aprašymas** | Svečias gali registruotis sistemoje |
| **Naudotojas (-ai)** | Svečias |
| **Prieš-sąlygos** | Veikia internetinis ryšys |
| **Pagrindinis scenarijus** | 1. Svečias inicijuoja registraciją. 2. Sistema pateikia dialogo langą duomenų įvedimui. 3. Svečias įveda informaciją ir patvirtina įvedimą 4. Sistema patvirtina informaciją. |
| **Alternatyvūs scenarijai** | 4.a Toks klientas jau yra sąraše.  4.a.1 Sistema informuoja naudotoją.  4.a.2 Naudotojas pakeičia informaciją.  4.a.3 Sistema patvirtina informaciją |
| **Po-sąlygos** | Svečias gali prisijungti prie sistemos |
| **Papildoma informacija** | Informacija apie klientą:  vardas, elektrinis paštas, slaptažodis |

|  |  |
| --- | --- |
| **Pavadinimas** | Registruotis sistemoje |
| **ID** | UC03 |
| **Trumpas aprašymas** | Svečias ir registruotas naudotojas gali peržiūrėti blogus |
| **Naudotojas (-ai)** | Svečias ir registruotas naudotojas |
| **Prieš-sąlygos** | Veikia internetinis ryšys |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja blogą 2. Naudotojas pateikia blogą |
| **Po-sąlygos** | Svečias ir registruotas naudotojas gali peržiūrėti blogus |

|  |  |
| --- | --- |
| **Pavadinimas** | Keisti asmeninę informaciją |
| **ID** | UC04 |
| **Trumpas aprašymas** | Sistema turi leisti registruotiems naudotojams keisti asmeninę informaciją |
| **Naudotojas (-ai)** | Registruotoms naudotojams |
| **Prieš-sąlygos** | Veikia internetinis ryšys |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja pakeisti informaciją 2. Sistema pateikia dialogo langą duomenų įvedimui. 3. Naudotojas įveda informaciją ir patvirtina įvedimą 4. Sistema patvirtina informaciją. |
| **Alternatyvūs scenarijai** | 4.a Neteisinga informacija.  4.a.1 Sistema informuoja naudotoją.  4.a.2 Naudotojas pakeičia informaciją.  4.a.3 Sistema patvirtina informaciją |
| **Po-sąlygos** | Pakeista informacija |
| **Papildoma informacija** | Informacija apie klientą:  elektrinis paštas, slaptažodis, vardas |

|  |  |
| --- | --- |
| **Pavadinimas** | Sukurti blogus |
| **ID** | UC05 |
| **Trumpas aprašymas** | Sistema turi leisti registruotoms naudotojams sukurti blogus |
| **Naudotojas (-ai)** | Registruotoms naudotojams |
| **Prieš-sąlygos** | Veikia internetinis ryšys |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja sukurti blogą 2. Sistema pateikia dialogo langą duomenų įvedimui. 3. Naudotojas įveda informaciją ir patvirtina įvedimą 4. Sistema patvirtina informaciją. |
| **Alternatyvūs scenarijai** | 4.a Neteisinga informacija.  4.a.1 Sistema informuoja naudotoją.  4.a.2 Naudotojas pakeičia informaciją.  4.a.3 Sistema patvirtina informaciją |
| **Po-sąlygos** | Sukurtas blogas |
| **Papildoma informacija** | Informacija apie blogą:  Tekstas ir nuotrauka |

|  |  |
| --- | --- |
| **Pavadinimas** | Keisti blogus |
| **ID** | UC06 |
| **Trumpas aprašymas** | Sistema turi leisti registruotoms naudotojams keisti blogus |
| **Naudotojas (-ai)** | Registruotoms naudotojams |
| **Prieš-sąlygos** | Veikia internetinis ryšys |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja sukurti blogą 2. Sistema pateikia dialogo langą duomenų įvedimui. 3. Naudotojas įveda informaciją ir patvirtina įvedimą 4. Sistema patvirtina informaciją. |
| **Alternatyvūs scenarijai** | 4.a Neteisinga informacija.  4.a.1 Sistema informuoja naudotoją.  4.a.2 Naudotojas pakeičia informaciją.  4.a.3 Sistema patvirtina informaciją |
| **Po-sąlygos** | Pakeistas blogas |
| **Papildoma informacija** | Informacija apie blogą:  Tekstas ir nuotrauka |

|  |  |
| --- | --- |
| **Pavadinimas** | Pašalinti naudotojus |
| **ID** | UC07 |
| **Trumpas aprašymas** | Sistema turi leisti administratoriui pašalinti naudotojus |
| **Naudotojas (-ai)** | administratorius |
| **Prieš-sąlygos** | Veikia internetinis ryšys |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja pašalinti naudotojus 2. Sistema pateikia dialogo langą duomenų įvedimui. 3. Naudotojas įveda informaciją ir patvirtina įvedimą 4. Sistema patvirtina informaciją. |
| **Alternatyvūs scenarijai** | 4.a Neteisinga informacija.  4.a.1 Sistema informuoja naudotoja.  4.a.2 Naudotojas pakeičia informaciją.  4.a.3 Sistema patvirtina informaciją |
| **Po-sąlygos** | Pašalintas naudotojas |
| **Papildoma informacija** | Informacija apie naudotojus:  Vardas elektrinis paštas |

|  |  |
| --- | --- |
| **Pavadinimas** | Pašalinti blogus |
| **ID** | UC08 |
| **Trumpas aprašymas** | Sistema turi leisti administratorių pašalinti blogus |
| **Naudotojas (-ai)** | administratorius |
| **Prieš-sąlygos** | Veikia internetinis ryšys |
| **Pagrindinis scenarijus** | 1. Naudotojas inicijuoja pašalinti blogą 2. Sistema pateikia dialogo langą duomenų įvedimui. 3. Naudotojas įveda informaciją ir patvirtina įvedimą 4. Sistema patvirtina informaciją. |
| **Alternatyvūs scenarijai** | 4.a Neteisinga informacija.  4.a.1 Sistema informuoja naudotoja.  4.a.2 Naudotojas pakeičia informaciją.  4.a.3 Sistema patvirtina informaciją |
| **Po-sąlygos** | Pašalintas blogas |
| **Papildoma informacija** | Informacija apie blogą:  Tekstas ir pavadinimas |

* 1. NEFUNKCINIAI REIKALAVIMAI

Sistemos pagrindiniai nefunkciniai reikalavimai.

1. Sistema veikimo greitis sekundėmis, kito puslapio užkrovimui.
2. Kodo „templat“ pasikartojimas ir panaudojimas kitose vietose.
3. ARCHITEKTŪROS APRAŠYMAS

Sistemos architektūrai pasirinktas MVC (ang. Model View Controller) modelis. „Model“ tvarko sistemos duomenis ir su jais susijusias operacijas. „View“ tvarko duomenų rodymą naudotojams. „Controller“ tvarko naudotojų veiksmus ir siunčia juos į „View“ ir „Model“ komponentus.

Diagram

Description automatically generated

pav. MVC architektūra

1. PROGRAMŲ SISTEMOS PROJEKTINĖ DALIS
   1. FUNKCINIŲ REKALAVIMŲ TRASAVIMO LENTELĖ

|  |  |  |
| --- | --- | --- |
| **ID** | **FUNKCINIS REIKALAVIMAS** | **KOKIAME FUNKCINIAME SISTEMOS KOMPONENTE LOKALIZUOTAS** |
| FR1 | Prisijungti | Logika atliekama „Controler“ dalyje tikrina duomenis paimtus iš „Model“ dalyje paima duomenis iš duomenų bazės. „View“ viska parodo naudotojui. |
| FR2 | Registruotis | Logika atliekama „Controler“ dalyje tikrina duomenis ir siunčia į „Model“ dalį kuri duomenis patalpina į duomenų baze. „View“ viska parodo naudotojui. |
| FR3 | Blogo peržiūra | Logika atliekama „Controler“ dalyje tikrina duomenis paimtus iš „Model“ dalyje paima duomenis iš duomenų bazės. „View“ viska parodo naudotojui. |
| FR4 | Keisti informaciją | Logika atliekama „Controler“ dalyje tikrina duomenis ir siunčia į „Model“ dalį kuri duomenis patalpina į duomenų baze. „View“ viska parodo naudotojui. |
| FR5 | Kurti blogą | Logika atliekama „Controler“ dalyje tikrina duomenis paimtus iš „Model“ dalyje paima duomenis iš duomenų bazės. „View“ viska parodo naudotojui. |
| FR6 | Keisti blogą | Logika atliekama „Controler“ dalyje tikrina duomenis ir siunčia į „Model“ dalį kuri duomenis patalpina į duomenų baze. „View“ viska parodo naudotojui. |
| FR7 | Pašalinti naudotoją | Logika atliekama „Controler“ dalyje tikrina duomenis ir siunčia į „Model“ dalį kuri duomenis šalina iš duomenų baze. „View“ viska parodo naudotojui. |
| FR8 | Pašalinti blogą | Logika atliekama „Controler“ dalyje tikrina duomenis ir siunčia į „Model“ dalį kuri duomenis šalina iš duomenų baze. „View“ viska parodo naudotojui. |

* 1. DUOMENŲ BAZĖS MODELIS

Sudarytas duomenų bazės modelis esybėlių ryšių diagramą:

Diagram

Description automatically generated

pav. Esybių ryšių diagramą

Duomenų bazėje yra dvi lentelės. Naudotojai kurie turi ID, elektroninį paštą, naudotojo vardą, registravimosi data, ar yra administratorius. Blogas turi ID, išorinį raktą kuris yra blogą sukūrusio naudotojo ID, blogo titulas , pagrindinis tekstas, blogo nuotrauka, blogo sukūrimo data. Naudotojas gali turėti nulį arba daug blogų, bet blogas gali turėti tik viena naudotoją kuris jį sukūrė.

* 1. KLASIŲ DIAGRAMA / ANALIZĖS OBJEKTO DIAGRAMA

Sudaryta klasių diagrama:

Diagram

Description automatically generated

pav. Klasių diagrama

Iš blogo svetainės sudaryta klasių diagrama su klasėmis: „blog“, „account“.

* 1. SISTEMOS ELGSENOS DIAGRAMOS

Sistemos panaudos atvejų veiklos diagramos:

Diagram

Description automatically generated

pav. Veiklos diagrama prisijungimui

Diagram

Description automatically generated

pav. Veiklos diagrama registracijai

Diagram

Description automatically generated

pav. Veiklos diagrama blogo peržiūrai

Diagram

Description automatically generated

pav. Veiklos diagrama keisti informaciją

Diagram

Description automatically generated

pav. Veiklos diagrama keisti informaciją

Diagram

Description automatically generated

pav. Veiklos diagrama sukurti blogus

Diagram

Description automatically generated

pav. Veiklos diagrama keisti blogus

Diagram

Description automatically generated

pav. Veiklos diagrama pašalinti naudotoją

Diagram

Description automatically generated

pav. Veiklos diagrama pašalinti blogą

1. REALIZAVIMO ĮRANKIŲ IR PRIEMONIŲ APRAŠYMAS

Svetainei panaudota „Python“ programavimo kalba. Tinklalapio „backend“ naudotas karkasas „Django“. Tai pat „Frontend“ buvo panaudotas „Bootstrap“ karkasas pagražinti tinklalapį. Veido suradimui ir atpažinimui panaudota biblioteka „OpenCV“.

Text

Description automatically generated

pav. Failas „requirements.txt “

Tai pat projekte esančiame faile „requirements.txt “ tiksliau išvardintos papildomos naudotos bibliotekos. (pav. 20)

1. TESTAVIMO PLANAS

Sistemos testavimas atliekamas dviem lygiais. Pirmas lygis atliekamas su „Unit“ testais parašytais tinklalapiui testuoti. Antram lygiui bus naudojama sistemos testavimas iš naudotojo pusės parašytais scenarijais.

1. NAUDOTOJO INSTRUKCIJA
2. Programą galima paleisti naudojantis „Visual Studio“ įrankiu, atidarius aplanką „src“ (pav. 2).

A picture containing text, screenshot, monitor

Description automatically generated

pav. Programos aplankalo atidarymas

1. Atidarius programą šone rodo projekto aplikacijų aplankalus ir „manage.py“ programą. Norint paleisti programą, reikia viršuje pasirinkti „Terminal/New Terminal“ (pav. 3).

Graphical user interface

Description automatically generated

pav. Terminalo paleidimas

1. Norint įjungti terminalą reikia įrašyti komandą „python manage.py runserver“ ir paspausti „Enter“ mygtuką (pav. 4).

A screenshot of a computer

Description automatically generated with medium confidence

pav. Terminalo komanda

1. Paspaudus „Enter“ atsiras nuoroda į tinklalapį „http://127.0.0.1:8000/“ (pav. 5)

Graphical user interface, text, website

Description automatically generated

pav. Nuoroda į tinklalapį

1. Pagal nuorodą naršyklėje atsidarys projekto tinklalapis (pav. 6).

Graphical user interface, application, Teams

Description automatically generated

pav. Svetainė naršyklėje

1. Toliau galima svetainėje registruotis, prisijungti, peržiūrėti sukurtus blogus ir juos kurti.

IŠVADOS

Buvo sukurta svetainė naudojant „python“ programavimo kalbą, „Django“ ir „Bootstrap“ karkasus. Projekte aprašytas projekto planas, funkciniai ir nefunkciniai reikalavimai, pristatyta architektūra ir naudotojo instrukcijos.