

INFORMATIKOS FAKULTETAS

**T120B165 Saityno taikomųjų programų projektavimas**  
**Projekto ataskaita**

Studentas: Vytenis Kriščiūnas, IFF-1/1

Dėstytojai: lekt. Kiudys Eligijus  
lekt. prakt. Baltulionis Simonas

KAUNAS, 2024

# TURINYS

1.	Įvadas .....	4
2.	Uždavinio aprašymas .....	5
2.1.	Sistemos paskirtis.....	5
2.2.	Funkciniai reikalavimai .....	5
3.	Sistemos architektūra .....	7
4.	Naudotojo sąsajos projektas.....	8
5.	API specifikacija (yra atskiras .yaml failas) .....	18
6.	Išvados .....	21
7.	GitHub.....	22

## PAVEIKSLĖLIAI

1 pav. UML deployment diagrama .....	7
2 pav. Wireframe home page .....	8
3 pav. Home page lango iškarpa .....	8
4 pav. Wireframe Login .....	9
5 pav. Login page lango iškarpa .....	9
6 pav. Wireframe Register page .....	10
7 pav. Register page lango iškarpa.....	10
8 pav. Wireframe Subjects page.....	11
9 pav. Subjects page lango iškarpa .....	11
10 pav. Wireframe Create New Subject pop up .....	12
11 pav. Create New Subject pop up lango iškarpa.....	12
12 pav. Wireframe Subject page ir jo Tasks.....	13
13 pav. Subject page ir jo Tasks lango iškarpa .....	13
14 pav. Wireframe update Subject pop up .....	14
15 pav. Update Subject pop up lango iškarpa .....	14
16 pav. Wireframe Task page ir komentarų sekcija .....	15
17 pav. Task page ir komentarų sekcijos lango iškarpa .....	15
18 pav. Wireframe Comment page.....	16
19 pav. Comment page lango iškarpa .....	16
20 pav. Wireframe Edit Comment pop up .....	17
21 pav. Edit Comment pop up lango iškarpa .....	17
22 pav. OpenAPI specifikacijos Temos API iškvietai .....	18
23 pav. OpenAPI specifikacijos Užduoties API iškvietai .....	19
24 pav. OpenAPI specifikacijos Komentarų API iškvietai .....	19
25 pav. OpenAPI specifikacijos autentifikacijos API iškvietai .....	20

# 1. Įvadas

Šios ataskaitos tikslas yra dokumentuoti gautus galutinio projekto rezultatus.

Uždaviniai:

- Sprendžiamo uždavinio aprašymas:
  - Sistemos paskirtis,
  - Funkciniai reikalavimai;
- Sistemos architektūra (pakanka UML deployment diagramos);
- Naudotojo sąsajos projektas (projektuojamos sąsajos langų wireframe'ai ir juos atitinkančios realizacijos langų iškarpos);
- API specifikacija - struktūriškai aprašyti API naudojant OpenAPI (atskiras api-spec.yaml failas). Taip pat kiekvienam API metodui pateikti:
  - galimus atsako kodus (response code),
  - panaudojimo pavyzdžius (užklausa, atsakymas);
- Pateikti projekto išvadas;
- Ataskaita turi būti pateikiama projektui sukurtame wiki arba projekto kodo Git saugykloje (.readme).

## 2. Uždavinio aprašymas

### 2.1. Sistemos paskirtis

Sukurta internetinė moksleivių mokymosi svetainė OSLS (Online Student Learning Site). Jos paskirtis yra suteikti mokytojams galimybę talpinti mokomąją medžiagą internetinėje erdvėje ir kad moksleiviai galėtų iš jos mokytis. Svetainė leidžia mokytojams ir moksleiviams komentuoti užduotis, taip palengvinant komunikaciją ir mokymosi procesą.

### 2.2. Funkciniai reikalavimai

1. Taikomosios srities objektai ir jų hierarchinis ryšys.

Tema -> užduotis -> komentaras.

2. API metodai.

Tema.

- Gauti visų temų sąrašą.
- Sukurti naują temą.
- Gauti konkrečią temą pagal ID.
- Atnaujinti esamą temą.
- Ištrinti temą.

Užduotis.

- Gauti visų užduočių sąrašą.
- Sukurti naują užduotį.
- Gauti konkrečią užduotį pagal ID.
- Atnaujinti esamą užduotį.
- Ištrinti užduotį.

Komentaras

- Gauti visų komentarų sąrašą.
- Sukurti naują komentarą.
- Gauti konkretų komentarą pagal ID.
- Atnaujinti esamą komentarą.
- Ištrinti komentarą.

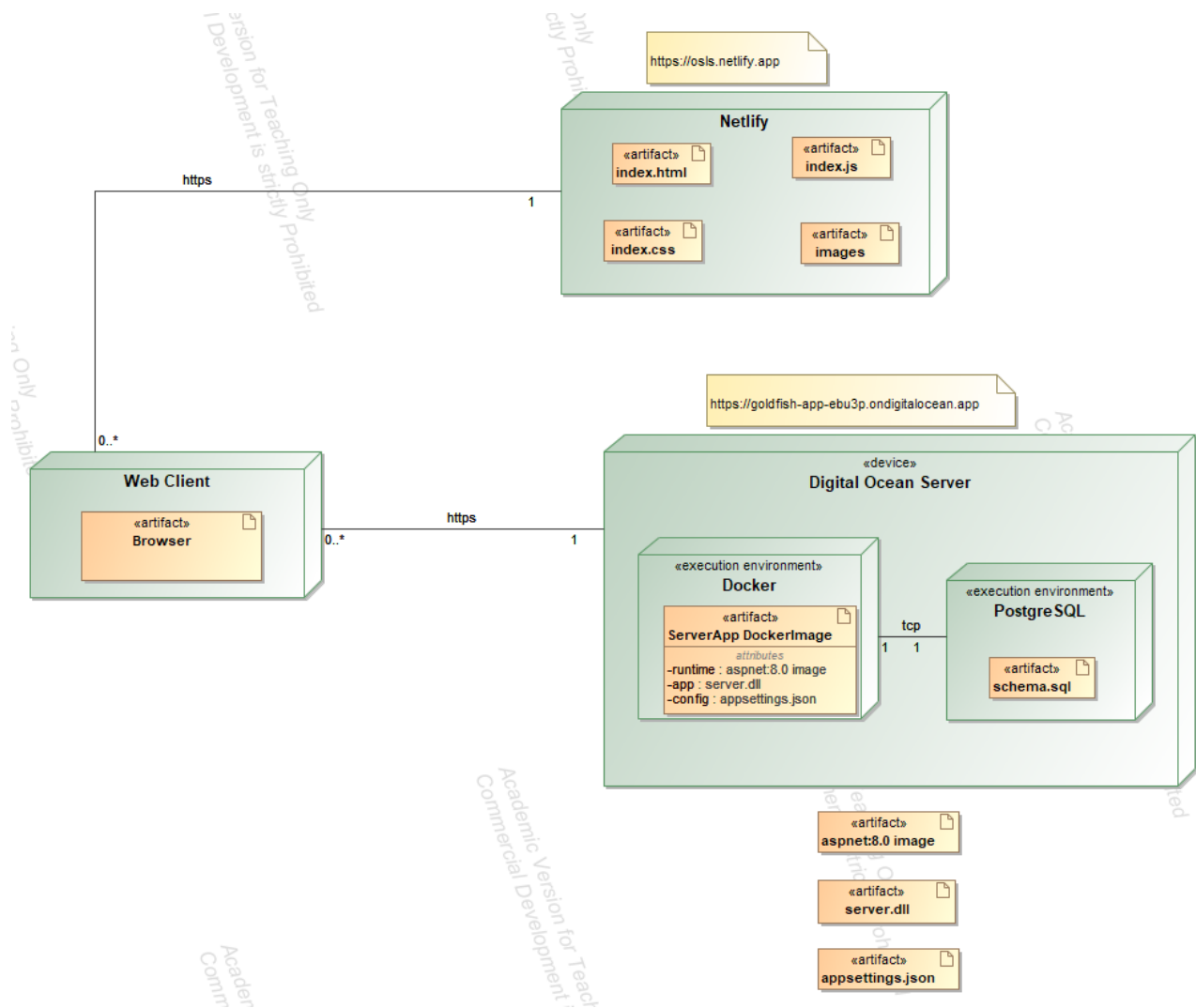
### 3. Hierarchinis API metodas.

Gauti visų konkrečios temos užduočių sąrašą.

### 4. Rolės.

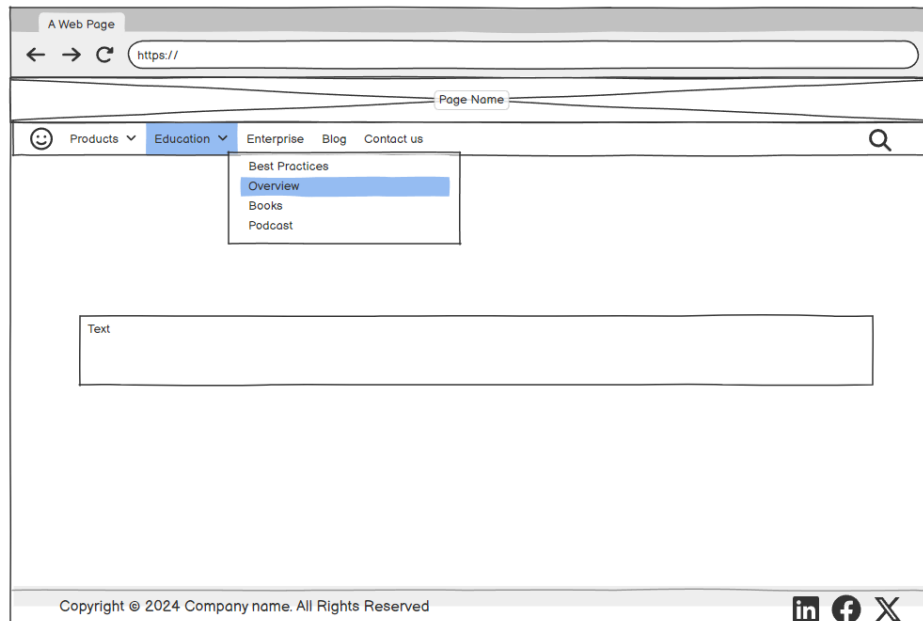
- **Svečias:** gali peržiūrėti temas ir užduotis, bet negali jų komentuoti.
- **Moksleivis:** gali peržiūrėti temas ir užduotis bei jas komentuoti.
- **Mokytojas:** gali peržiūrėti, kurti, redaguoti, trinti temas ir užduotis bei jas komentuoti.
- **Administratorius:** turi visas teises. Gali kurti, redaguoti, trinti temas ir užduotis bei valdyti vartotojų paskyras.

### 3. Sistemos architektūra

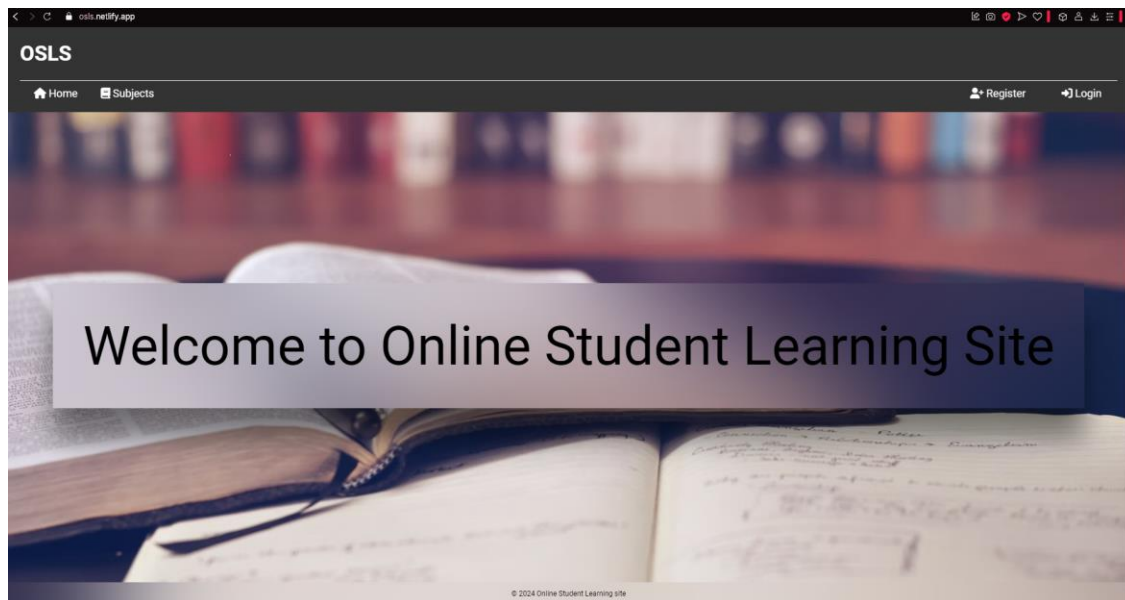


1 pav. UML deployment diagrama

## 4. Naudotojo sąsajos projektas

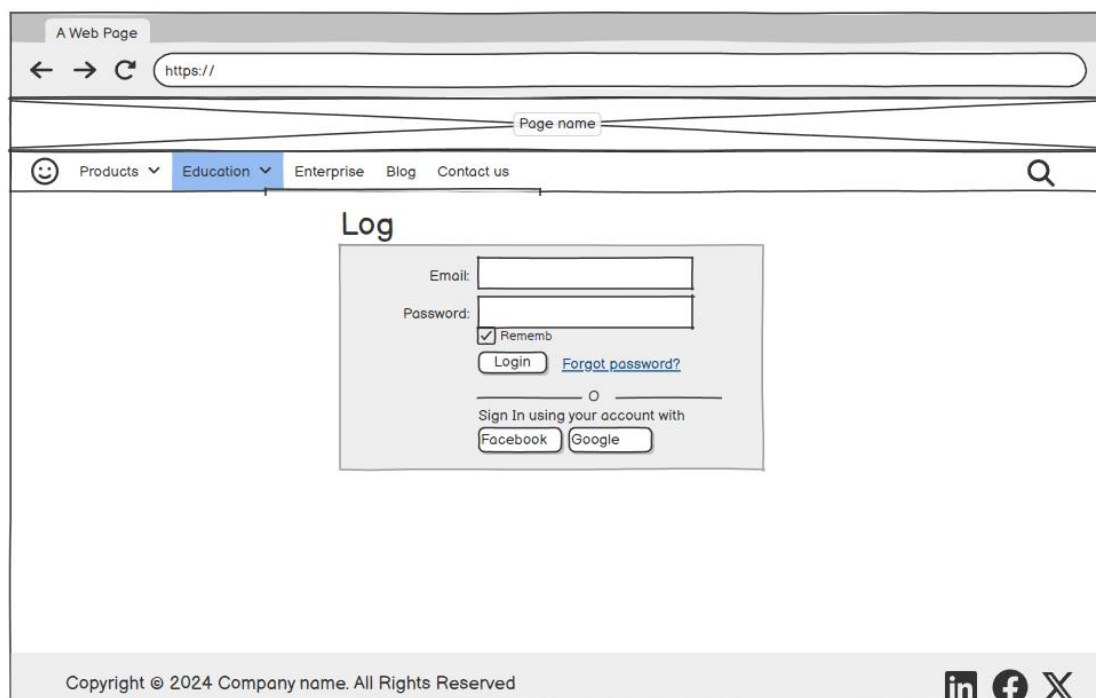


2 pav. Wireframe home page

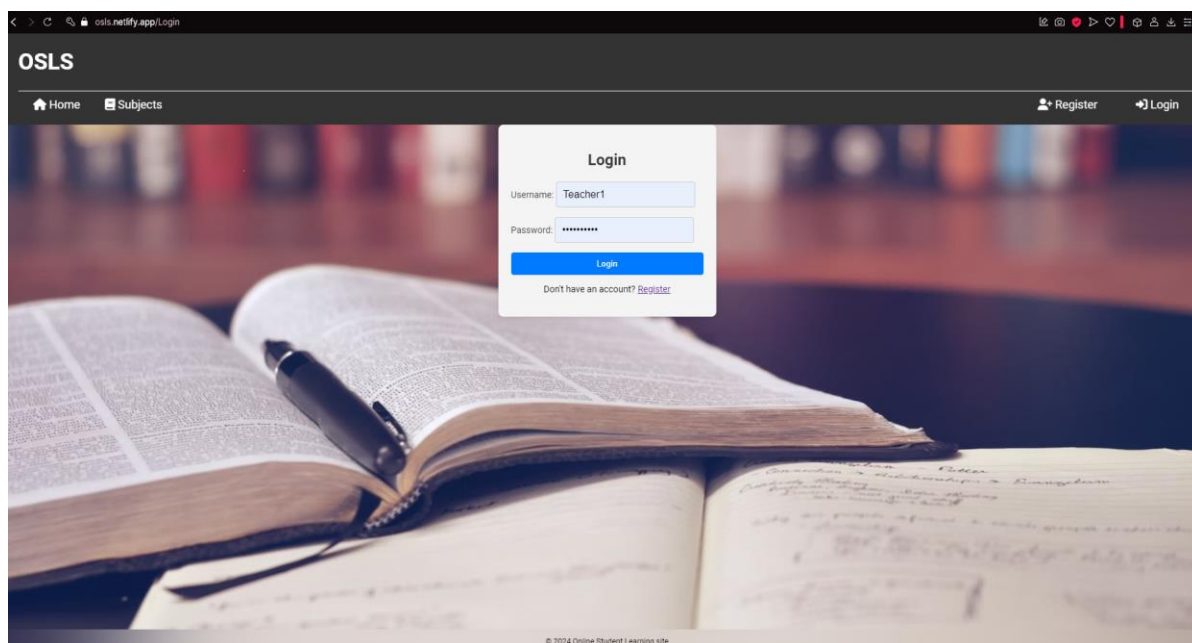


3 pav. Home page lango iškarpa

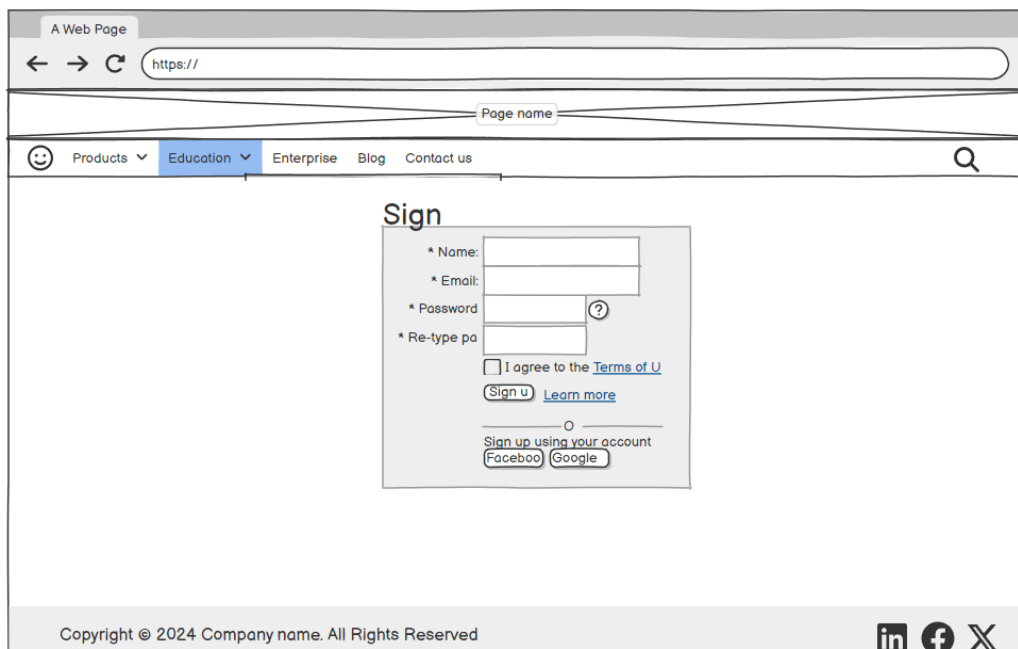




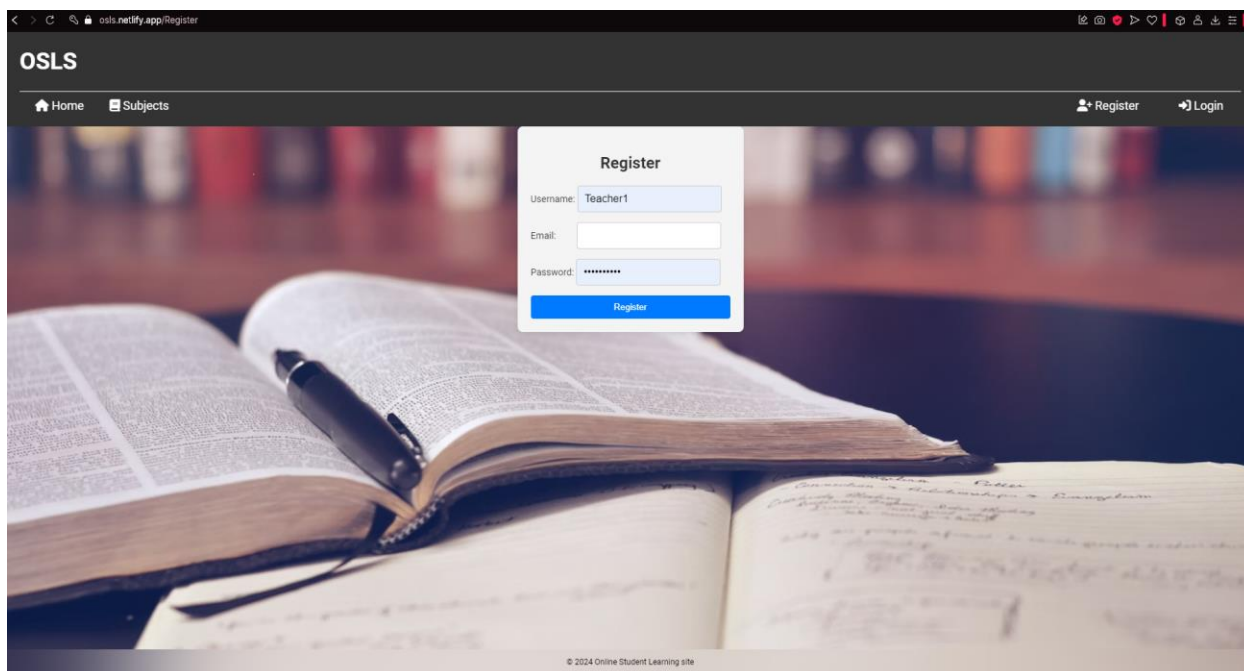
4 pav. Wireframe Login



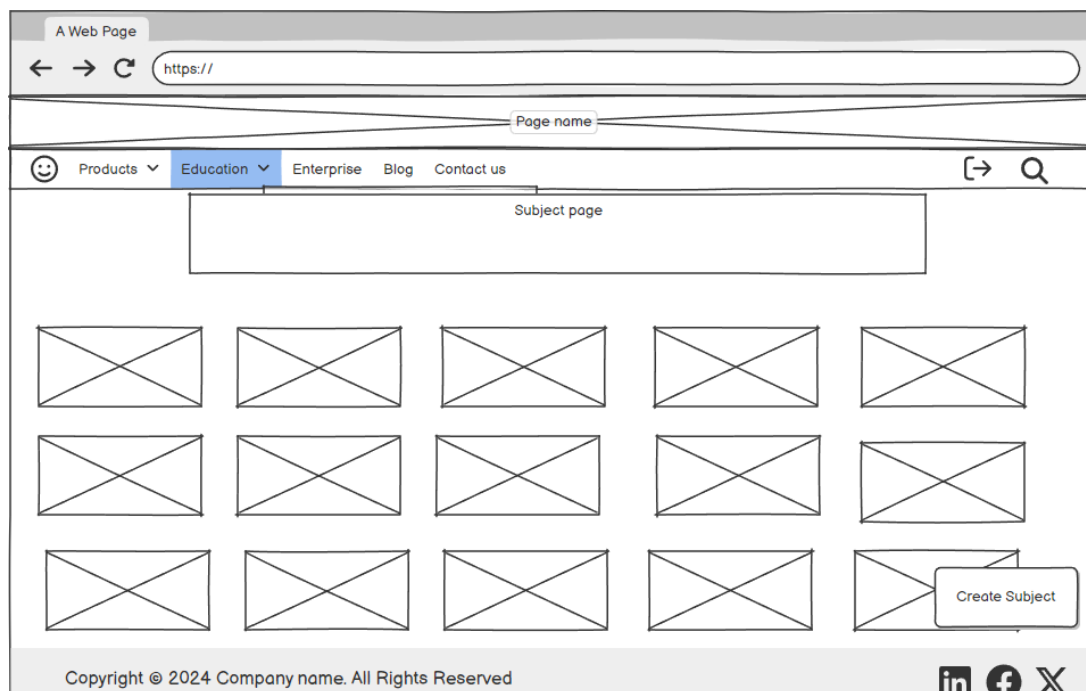
5 pav. Login page lango iškarpa



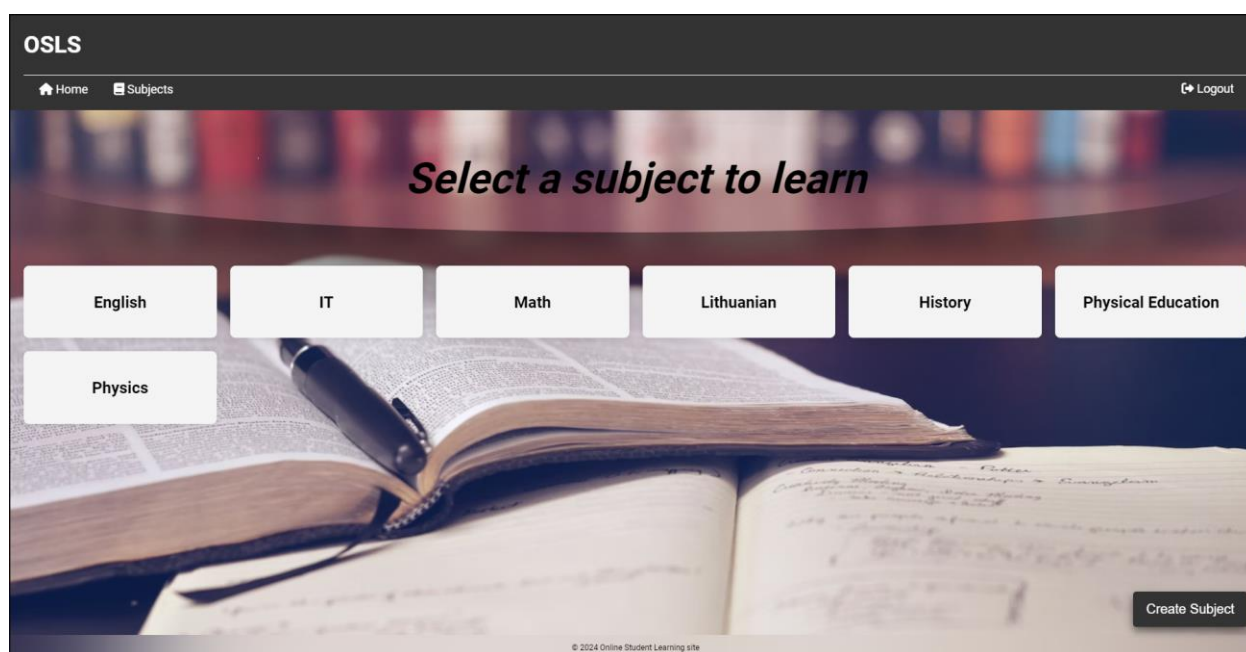
6 pav. Wireframe Register page



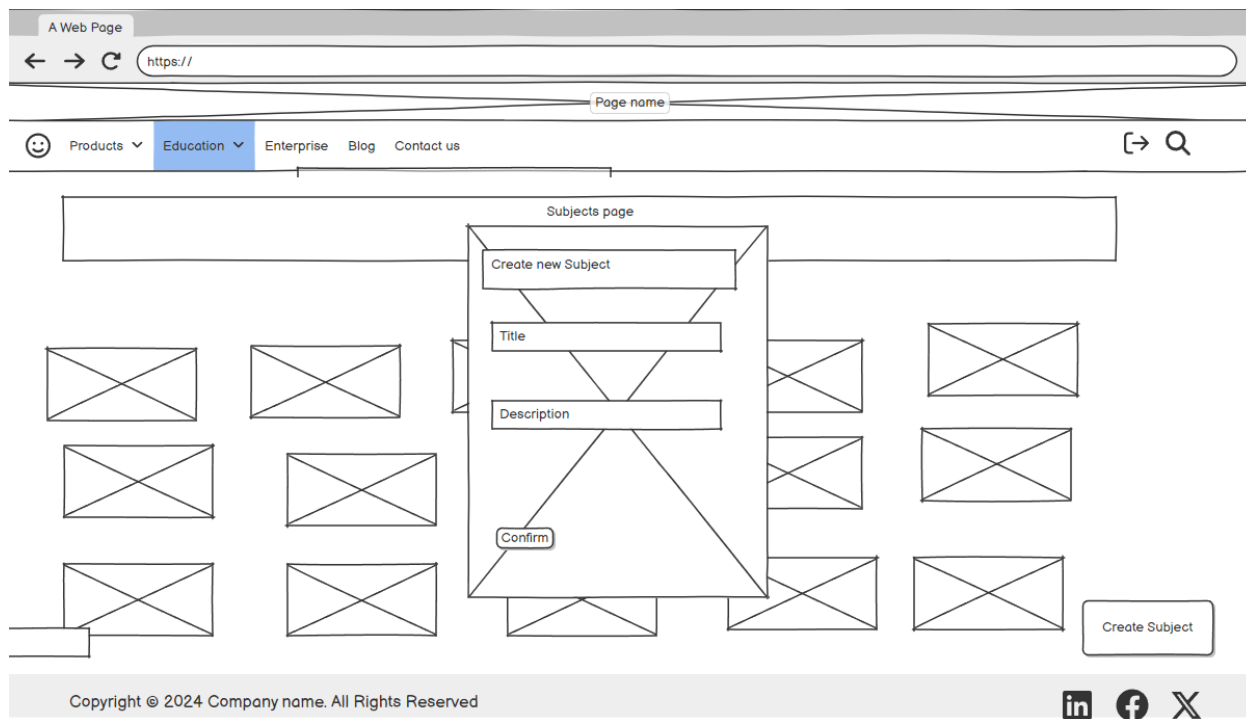
7 pav. Register page lango iškarpa



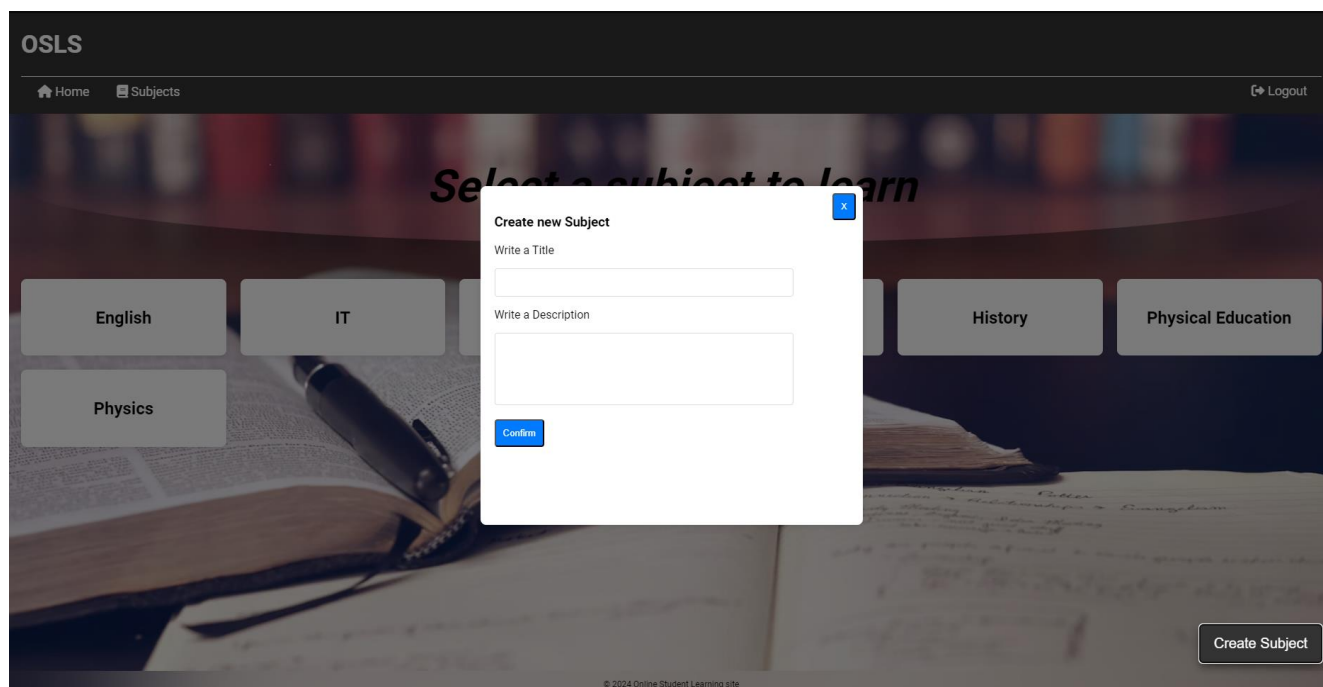
8 pav. Wireframe Subjects page



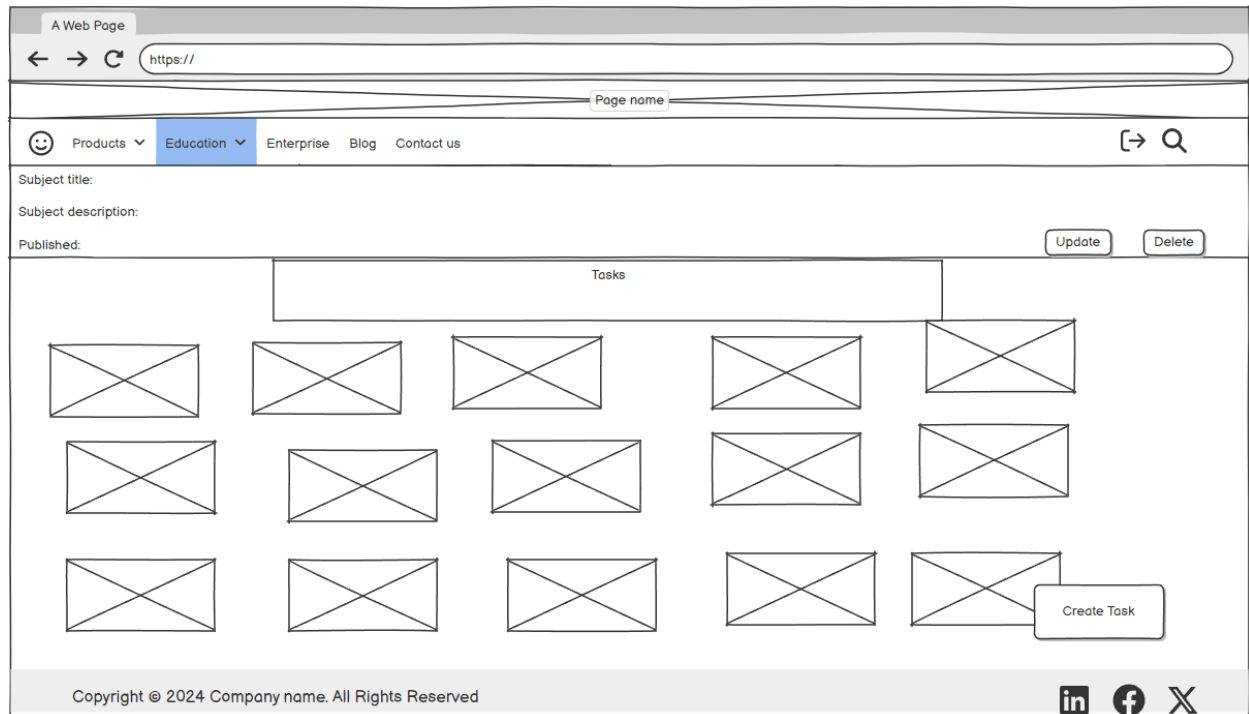
9 pav. Subjects page langų iškarpa



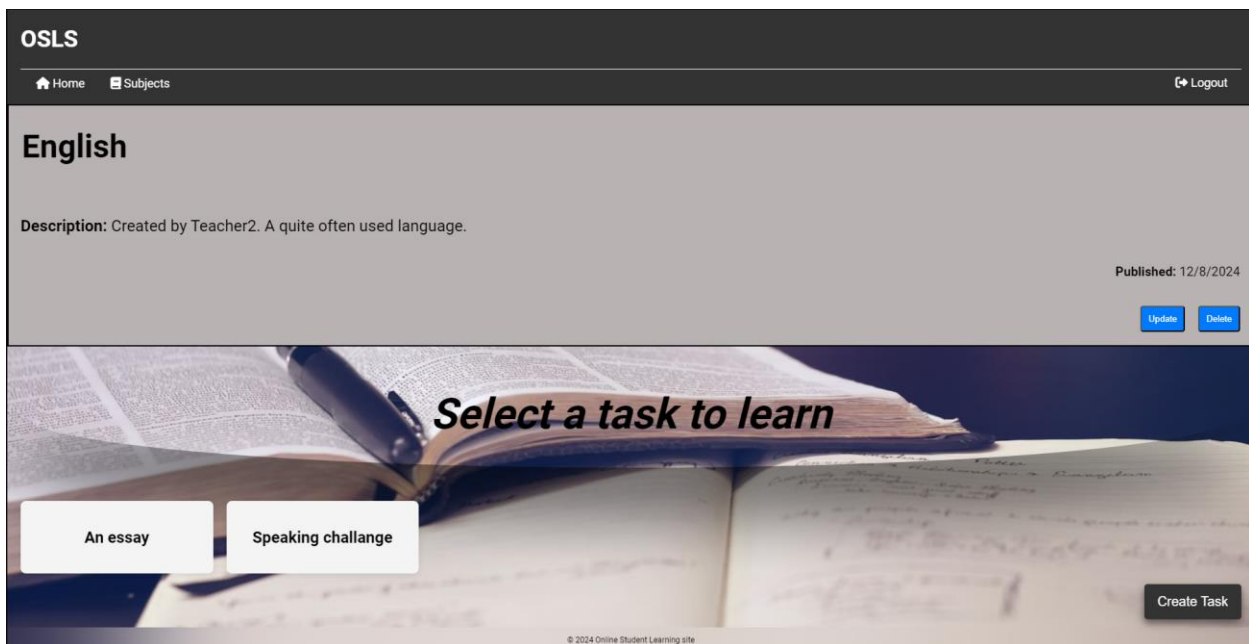
10 pav. Wireframe Create New Subject pop up



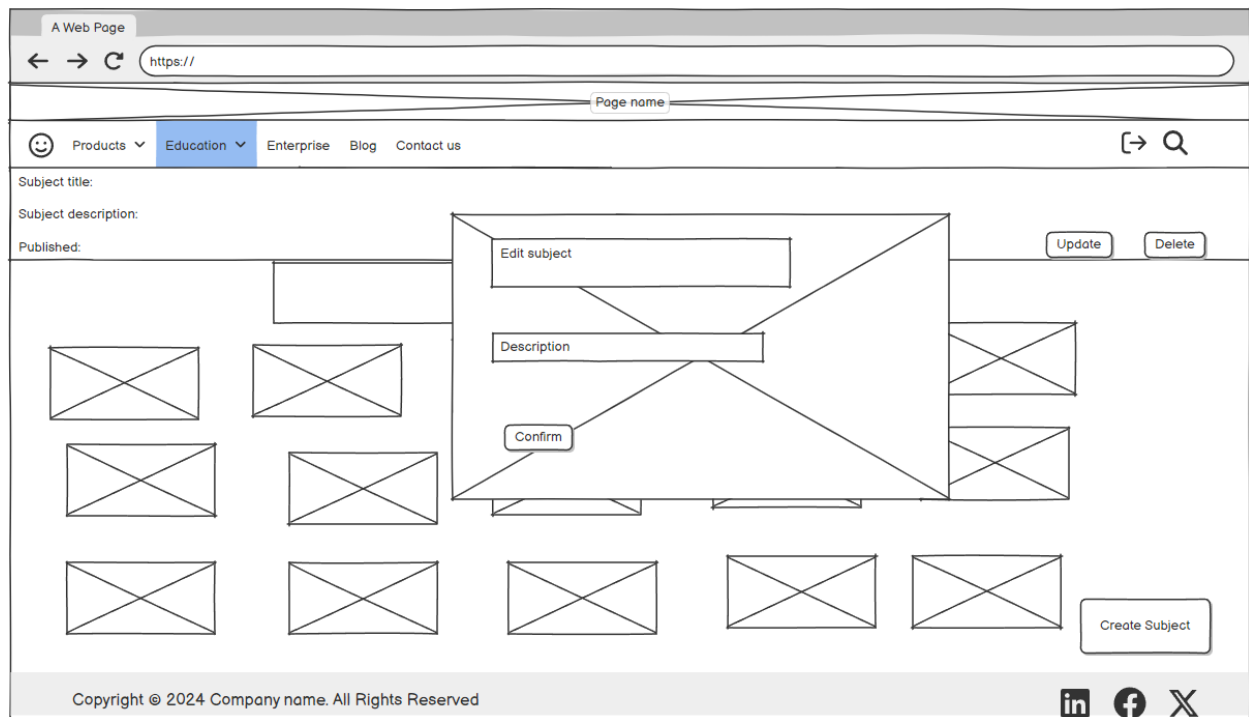
11 pav. Create New Subject pop up langa iškarpa



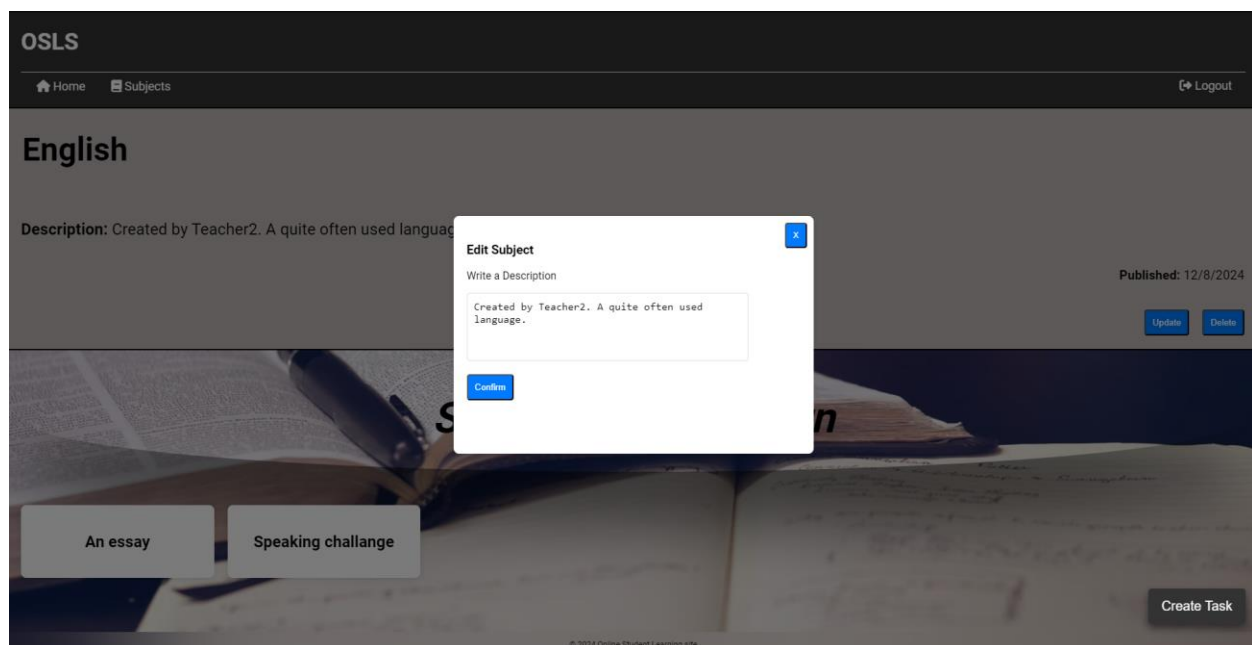
12 pav. Wireframe Subject page ir jo Tasks



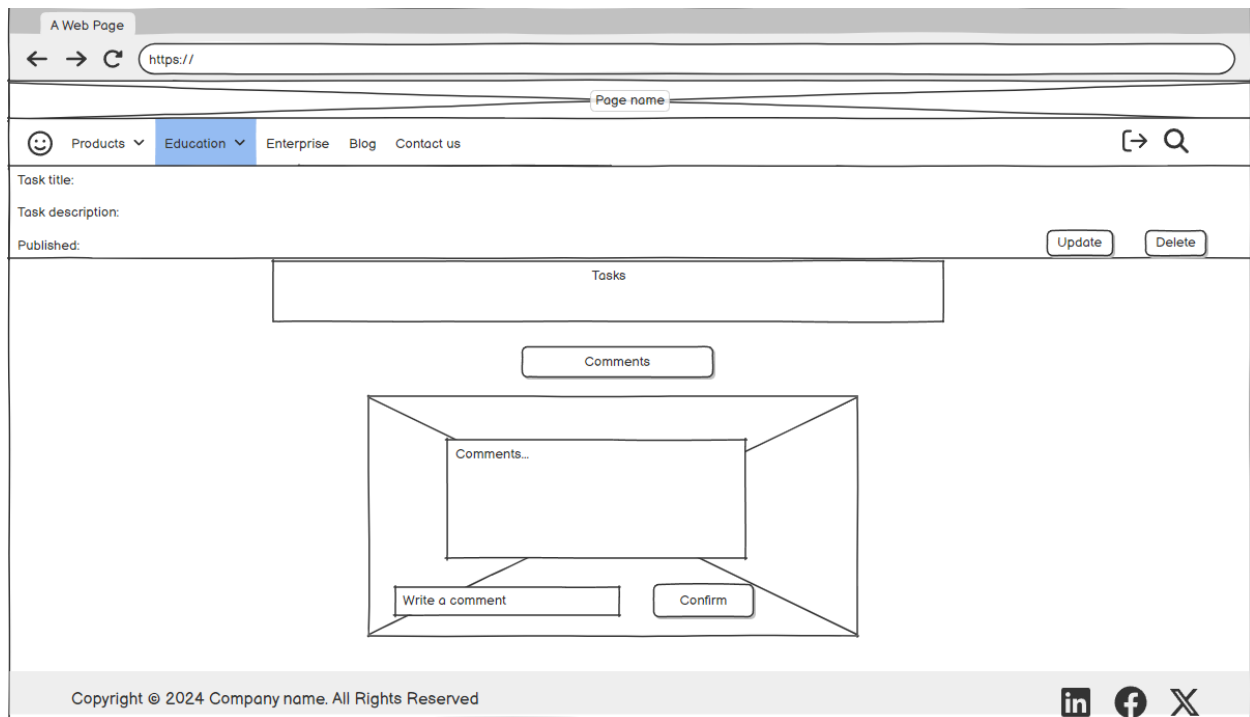
13 pav. Subject page ir jo Tasks lango iškarpa



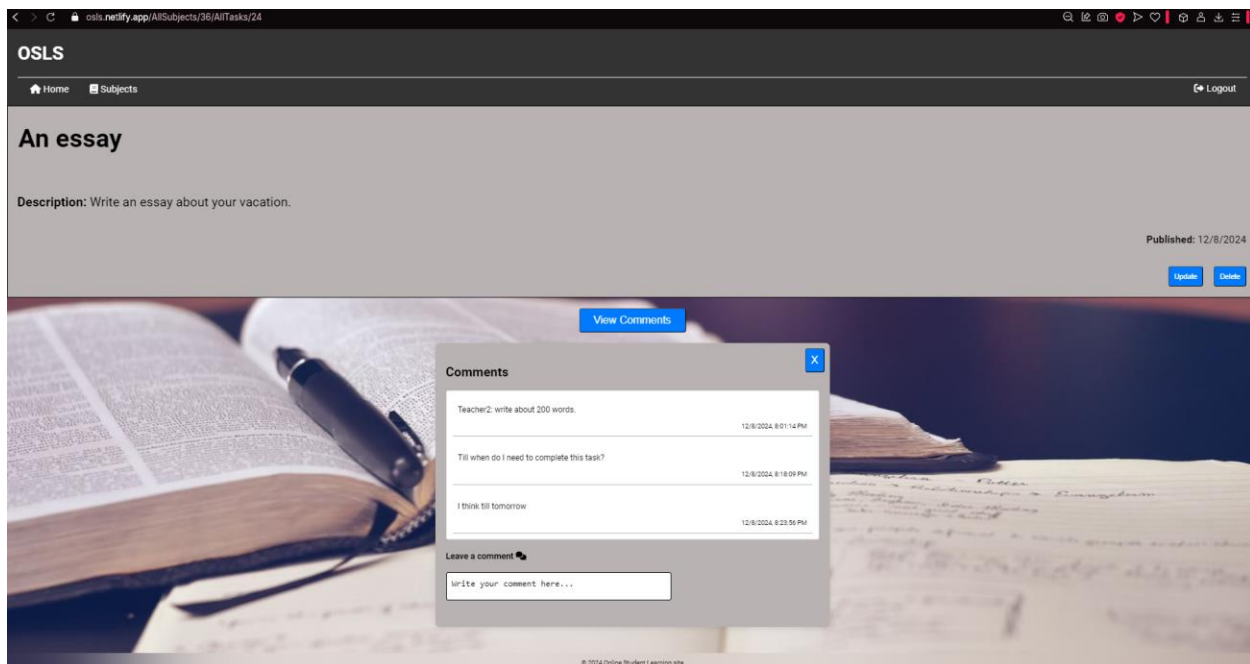
14 pav. Wireframe update Subject pop up



15 pav. Update Subject pop up lango iškarpa

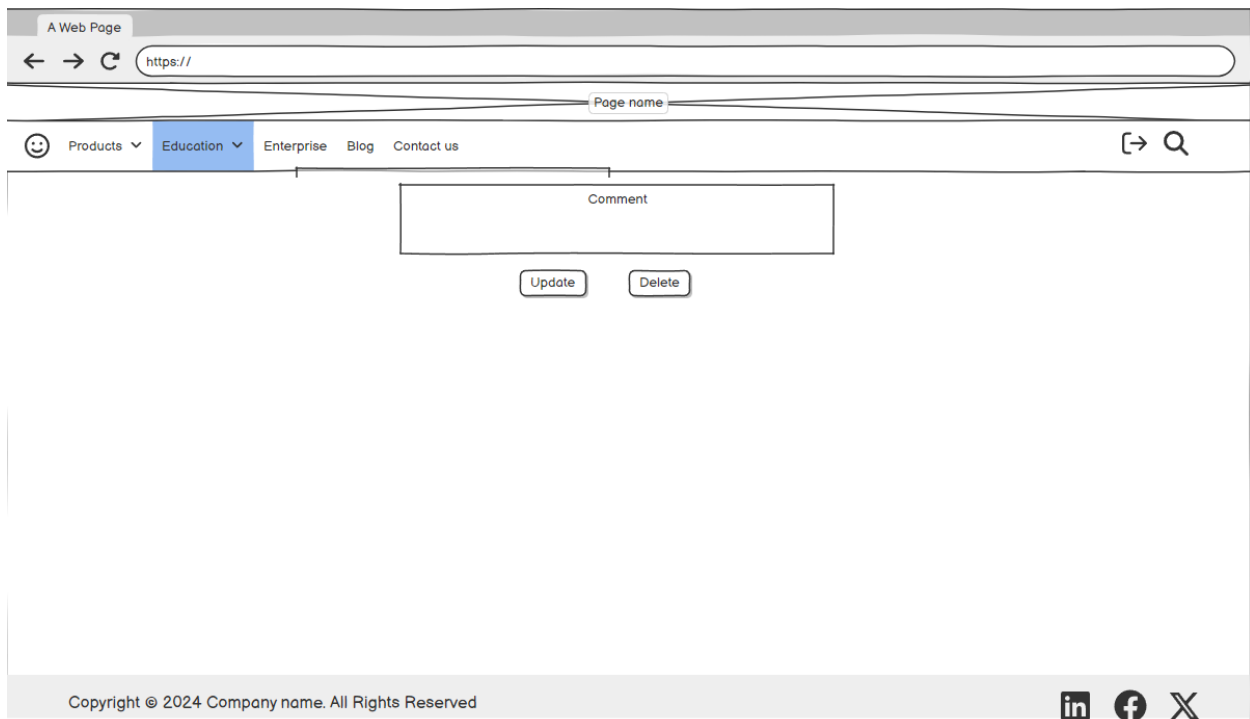


16 pav. Wireframe Task page ir komentų sekcija

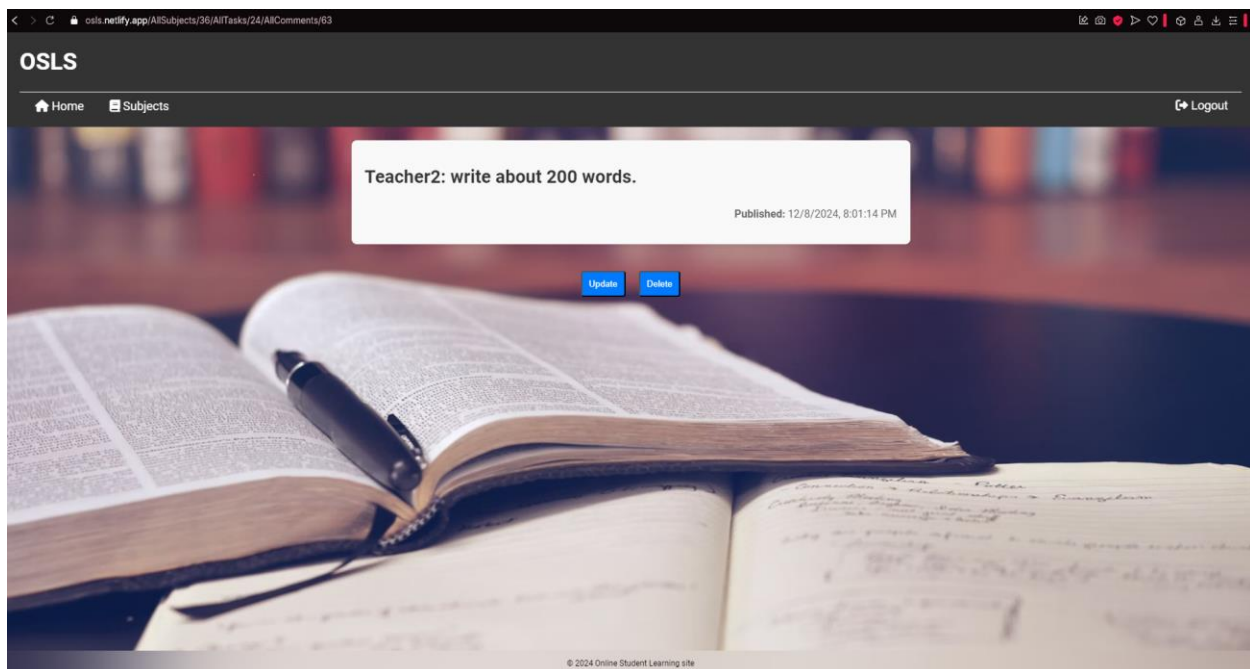


17 pav. Task page ir komentų sekcijos lango iškarpa



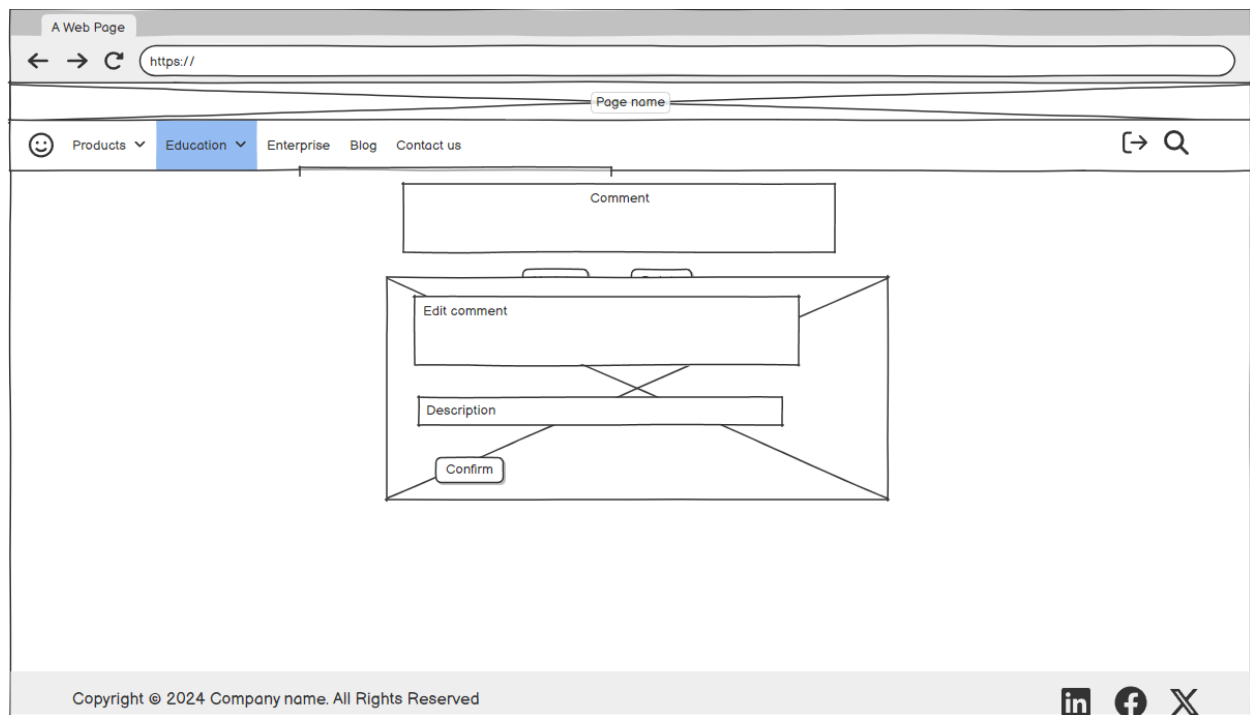


18 pav. Wireframe Comment page

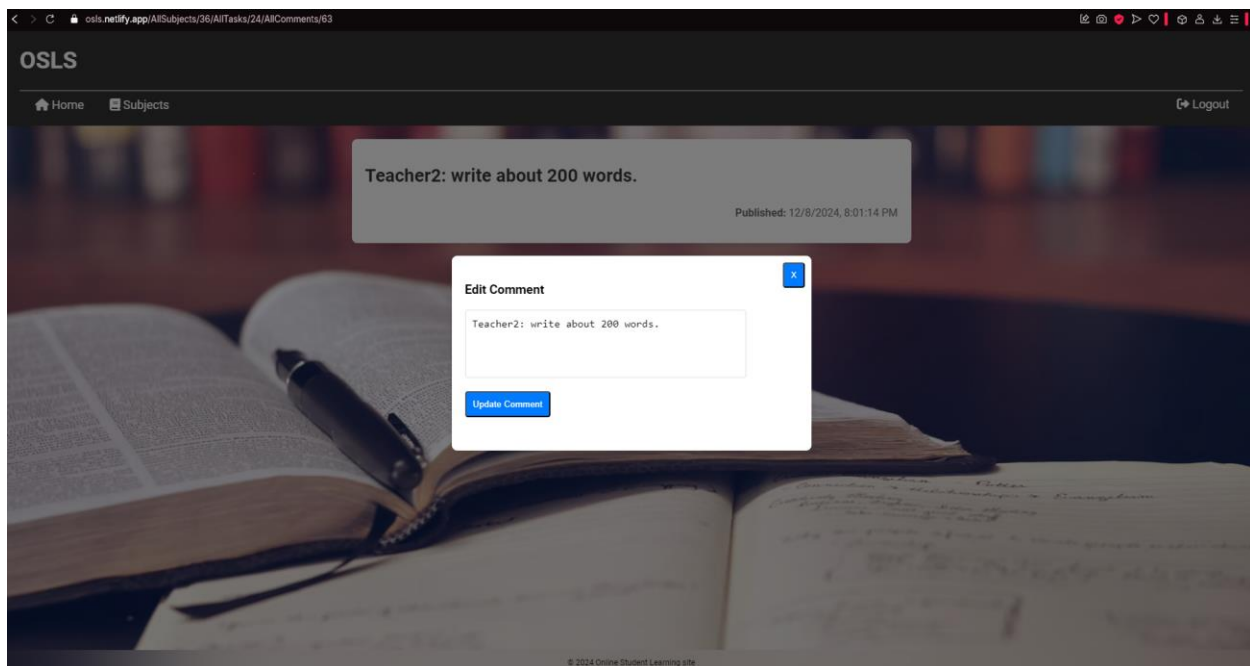


19 pav. Comment page lango iškarpa





20 pav. Wireframe Edit Comment pop up



21 pav. Edit Comment pop up lango iškarpa

## 5. API specifikacija (yra atskiras .yaml failas)

**API Title** 1.0 OAS 3.0

API's of subject, task and comment

Servers

`https://api.server.test` ▾

---

**subject** Operation related to subjects ^

GET	/subjects	get list of subjects	▾
POST	/subjects		▾
GET	/subjects/{subjectId}	Get a subject by ID	▾
PUT	/subjects/{subjectId}	Update a subject by ID	▾
DELETE	/subjects/{subjectId}	Delete a subject by ID	▾

22 pav. OpenAPI specifikacijos Temos API iškvietimai

## task Operations related to tasks



GET	/subjects/{subjectId}/tasks	Get tasks for a subject	▼
POST	/subjects/{subjectId}/tasks		▼
GET	/subjects/{subjectId}/tasks/{taskId}	Get a specific task for a subject	▼
PUT	/subjects/{subjectId}/tasks/{taskId}	Update a task by ID	▼
DELETE	/subjects/{subjectId}/tasks/{taskId}	Delete a task by ID	▼

23 pav. OpenAPI specifikacijos Užduoties API iškvietimai

## comment Operations related to comments



GET	/subjects/{subjectId}/tasks/{taskId}/comments	Get comments for a specific task	▼
POST	/subjects/{subjectId}/tasks/{taskId}/comments		▼
GET	/subjects/{subjectId}/tasks/{taskId}/comments/{commentId}	Get a specific comment	▼
PUT	/subjects/{subjectId}/tasks/{taskId}/comments/{commentId}	Update a comment by ID	▼
DELETE	/subjects/{subjectId}/tasks/{taskId}/comments/{commentId}	Delete a comment by ID	▼

24 pav. OpenAPI specifikacijos Komentaro API iškvietimai

<b>login</b> Operation related to login		^
POST	/login login a user	▼
<b>register</b> Operation related to register		^
POST	/accounts Register a user	▼
<b>logout</b> Operation related to logout		^
POST	/logout Log out a user	▼
<b>newAccessToken</b> Operation related to getting new access token		^
POST	/accessToken Renew access token using a refresh token	▼

25 pav. OpenAPI specifikacijos autentifikacijos API iškvietimai

## 6. Išvados

- 1) Dokumentuota ir aprašyta sukurta sistema;
- 2) Sukurta OpenAPI specifikacija visiems penkiolikai API iškvietimų;
- 3) Realizuota .NET serverio projekto pusė su penkiolika API iškvietimų, autentifikacija ir autorizacija;
- 4) Realizuota React kliento projekto pusė su visai reikalingais naudotojo sąsajos aspektais;
- 5) Projekto kliento ir serverio dalys patalpintos atskirose debesijos platformose;
- 6) Sukurta galutinė projekto ataskaita.

## 7. GitHub

Nuoroda į GitHub: <https://github.com/vyteniskri/IMMS>