



INFORMATIKOS FAKULTETAS

T120B165 Saityno taikomųjų programų projektavimas Projekto „Solea“ ataskaita

Studentas:

Liudas Baublys, IFF-0/3

Dėstytojai:

Doc. Tomas Blažauskas

KAUNAS 2023

Turinys

1. Sprendžiamo uždavinio aprašymas.....	3
1.1. Sistemos paskirtis.....	3
1.2. Funkciniai reikalavimai.....	3
2. Sistemos architektūra.....	4
3. Github nuoroda	4
4. SQL dumpas.....	4
5. API specifikacija.....	5
5.1. Naudotojų API metodų lentelė:.....	5
5.2. Miestų API metodų lentelė:.....	6
5.3. Vietovių API metodų lentelė:	7
5.4. Komentarų API metodų lentelė:.....	8

1. Sprendžiamo uždavinio aprašymas

1.1. Sistemos paskirtis

Leisti vartotojams peržiūrėti užduotis klausimus ir atsakymus į tuos klausimus. O registruotiems vartotojams pasklebti klausimus, atsakymus, bei uždėti „like“ arba „dislike“.

1.2. Funkciniai reikalavimai

Taikomosios srities objektai tarpusavyje susieti prasminiu ir hierarchiniu ryšiu:

- Vartotojas: Pagrindinis objektas.
 - Klausimas: Sub-objektas, reprezentuojantis užduotus klausimus.
 - Atsakymas: Sub-objektas, reprezentuojantis atsakymus į klausimus

API Metodai:

Klausimas:

- Peržiūrėti visus klausimus.
- Sukurti naują klausimą.
- Peržiūrėti konkretaus klausimo informaciją.
- Atnaujinti klausimo informaciją.
- Ištrinti klausimą.

Atsakymas:

- Peržiūrėti visus atsakymus konkretaus klausimo.
- Sukurti naują atsakymą.
- Peržiūrėti konkretaus atsakymo informaciją.
- Atnaujinti atsakymo informaciją.
- Ištrinti atsakymą.

Vartotojas:

- Peržiūrėti visus vartotojus.
- Sukurti naują vartotoją.
- Peržiūrėti konkretaus vartotojo informaciją.
- Atnaujinti vartotoją.
- Ištrinti vartotoją.

Įvertinimas:

- Peržiūrėti visus įvertinimus.
- Sukurti naują įvertinimą klausimui arba atsakymui.
- Peržiūrėti konkretaus įvertinimo informaciją.
- Atnaujinti įvertinimą.
- Ištrinti įvertinimą.

Rolės:

Svečias galės:

- Aplankyti pagrindinį puslapį.
- Užsiregistruoti.
- Prisijungti naudodamas prisijungimo duomenis.
- Pamatyti klausimus ir jų atsakymus.

Narys galės:

- Pridėti naują klausimą ir atsakymą
- Įvertinti klausimus ir atsakymus
- Peržiūrėti, keisti ar trinti savo sukurtą klausimą arba atsakymą

Administratorius galės:

- Šalinti vartotojus
- Keisti, šalinti visus klausimus ir atsakymus.

2. Sistemos architektūra

Sistemos sudedamosios dalys:

- Klientinė dalis (*angl. front-end*) – .NET
- Serverinė dalis (*angl. back-end*) – Laravel
- Duomenų bazė – MySQL

3. Github nuoroda

<https://github.com/liuds3/Saitynai>

4. SQL dumpas

Git'e

5. API specifikacija

Sistamai buvo sukurta 23 API metodų (be authentication), iš viso turima 4 *Entity*, kurie yra:

- User
- Question
- Answer
- Like

5.1. Naudotojų API metodų lentelė:

HTTP Method	Purpose	Route	Request Body	Headers	Possible Response Bodies	Status Codes
GET	Retrieve all users	/api/users		Authorization: Bearer {token}	{'status': 200, 'users': [...]} or {'status': 404, 'message': "No records found"}	200, 404
GET	Retrieve a single user	/api/user/{id}		Authorization: Bearer {token}	{'status': 200, 'user': {...}} or {'status': 404, 'message': "No user found"}	200, 404
POST	Create a new user	/api/users	{'name': string, 'email': string, 'password': string, 'role': integer (optional), 'currency': integer (optional)}	Authorization: Bearer {token}, Content-Type: application/json	{'status': 200, 'message': "User created successfully"} or {'status': 422, 'errors': {...}}	200, 422
PUT	Update a user	/api/user/{id}	{'name': string, 'email': string, 'password': string, 'role': integer (optional), 'currency': integer (optional)}	Authorization: Bearer {token}, Content-Type: application/json	{'status': 200, 'message': "User updated successfully"} or {'status': 422, 'errors': {...}} or {'status': 404, 'message': "No user found"}	200, 422, 404
DELETE	Delete a user	/api/user/{id}		Authorization: Bearer {token}	{'status': 200, 'message': "User deleted successfully"} or {'status': 404, 'message': "No user found"}	200, 404

5.2.Klausimų API metodų lentelė:

HTTP Method	Purpose	Route	Request Body	Headers	Possible Response Bodies	Status Codes
GET	Retrieve all questions	/api/questions		Authorization: Bearer {token}	{'status': 200, 'questions': [...]} or {'status': 404, 'message': "No records found"}	200, 404
GET	Retrieve a single question	/api/question/{id}		Authorization: Bearer {token}	{'status': 200, 'question': {...}} or {'status': 404, 'message': "No question found"}	200, 404
POST	Create a new question	/api/questions	{'question': string, 'content!': string, 'user': string })	Authorization: Bearer {token}, Content-Type: application/json	{'status': 200, 'message': "Question created successfully"} or {'status': 422, 'errors': {...}}	200, 422
PUT	Update a question	/api/question/{id}	{'question': string, 'content!': string, 'user': string })	Authorization: Bearer {token}, Content-Type: application/json	{'status': 200, 'message': "Question updated successfully"} or {'status': 422, 'errors': {...}} or {'status': 404, 'message': "No question found"}	200, 422, 404
DELETE	Delete a question	/api/question/{id}		Authorization: Bearer {token}	{'status': 200, 'message': "Question deleted successfully"} or {'status': 404, 'message': "No question found"}	200, 404

5.3. Atsakymų API metodų lentelė:

HTTP Method	Purpose	Route	Request Body	Headers	Possible Response Bodies	Status Codes
GET	Retrieve all answers	/api/answers		Authorization: Bearer {token}	{'status': 200, 'answers': [...]} or {'status': 404, 'message': "No records found"}	200, 404
GET	Retrieve a single answer	/api/answer/{id}		Authorization: Bearer {token}	{'status': 200, 'answer': {...}} or {'status': 404, 'message': "No answer found"}	200, 404
POST	Create a new answer	/api/answers	{'question': string, 'content!': string, 'user': string })	Authorization: Bearer {token}, Content-Type: application/json	{'status': 200, 'message': "Answer created successfully"} or {'status': 422, 'errors': {...}}	200, 422
PUT	Update a answer	/api/answer/{id}	{'question': string, 'content!': string, 'user': string })	Authorization: Bearer {token}, Content-Type: application/json	{'status': 200, 'message': "Answer updated successfully"} or {'status': 422, 'errors': {...}} or {'status': 404, 'message': "No answer found"}	200, 422, 404
DELETE	Delete a answer	/api/answer/{id}		Authorization: Bearer {token}	{'status': 200, 'message': "Answer deleted successfully"} or {'status': 404, 'message': "No answer found"}	200, 404
GET	Get all answers by question id	/api/answers/{id}		Authorization: Bearer {token}	{'status': 200, 'answers': [...]} or {'status': 404, 'message': "No records found"}	200, 404

5.4.Įvertinimo API metodų lentelė:

HTTP Method	Purpose	Route	Request Body	Headers	Possible Response Bodies	Status Codes
GET	Retrieve all likes	/api/likes		Authorization: Bearer {token}	{'status': 200, 'likes': [...]} or {'status': 404, 'message': "No records found"}	200, 404
GET	Retrieve a single like	/api/like/{id}		Authorization: Bearer {token}	{'status': 200, 'like': {...}} or {'status': 404, 'message': "No likes found"}	200, 404
POST	Create a new like	/api/likes	{'question': string, 'content!': string, 'user': string }}	Authorization: Bearer {token}, Content-Type: application/json	{'status': 200, 'message': "Like created successfully"} or {'status': 422, 'errors': {...}}	200, 422
PUT	Update a like	/api/like/{id}	{'question': string, 'content!': string, 'user': string }}	Authorization: Bearer {token}, Content-Type: application/json	{'status': 200, 'message': "Like updated successfully"} or {'status': 422, 'errors': {...}} or {'status': 404, 'message': "No like found"}	200, 422, 404
DELETE	Delete a like	/api/like/{id}		Authorization: Bearer {token}	{'status': 200, 'message': "Like deleted successfully"} or {'status': 404, 'message': "No like found"}	200, 404
GET	Get all likes by question id	/api/likesQuestion/{id}		Authorization: Bearer {token}	{'status': 200, 'likes': [...]} or {'status': 404, 'message': "No likes found"}	200, 404

					"No records found"}	
GET	Get all likes by answer id	/api/likesAnswer/{id}		Authorization: Bearer {token}	{'status': 200, 'likes': [...]} or {'status': 404, 'message': "No records found"}	200, 404