

INFORMATIKOS FAKULTETAS

T120B165 Saityno taikomųjų programų projektavimas

Projekto „Auto Parts“ ataskaita

Studentas: Vilius Vainorius, IFF-9/8

Dėstytojai: doc. Rasa Mažutienė,
doc. Petras Tamošiūnas

KAUNAS 2022

Turinys

1. Sprendžiamo uždavinio aprašymas	3
1.1. Sistemos paskirtis	3
1.2. Funkciniai reikalavimai	3
2. Sistemos architektūra	4
3. API Specifikacija	6
3.1. CarParts API metodai	6
4. Svetainės Front-end įgyvendinimas	17
4.1. Main Menu langas	17
4.2. Login langas	18
4.3. Register langas	18
4.4. Automobilių langas	19
4.5. Automobilio kūrimo langas	19
4.6. Automobilio markės kūrimo langas	20
5. Išvados	21

1. Sprendžiamo uždavinio aprašymas

1.1. Sistemos paskirtis

Projekto tikslas – suteikti patogią, funkcionalią platformą žmonėms, norintiems rasti savo automobilio dalis, informaciją apie jas, ir nuorodas į tą dalį pardavinėjančią parduotuvę.

Veikimo principas – Prisijungę vartotojai gali talpinti automobilius, administratorius valdo visus vartotojus, prižiūri puslapio veiklą, redaguoja puslapius jei reikia. Neprisijungęs vartotojas gali žiūrėti skelbimus, matyti visa informaciją apie skelbimus, tačiau negali įdėti skelbimo.

Žmogus norintis įkelti naują automobilį ar dalį turi prisiregistruoti į puslapį, prisijungti, suvesti asmeninę informaciją, toliau gali sukurti skelbimą su pilna informacija ir įdėti skelbimą į puslapio skelbimų katalogą. Administratorius gali valdyti skelbimus, tikrinti juos, gali redaguoti puslapį. Neprisijungęs vartotojas gali tik lankytis po katalogus, matyti visus skelbimus, jų informaciją.

Brand -> Car -> CarPart

Automobilio markė -> Automobilis -> Automobilio dalis

1.2. Funkciniai reikalavimai

Neregistruotas sistemos naudotojas galės:

1. Peržiūrėti visus automobilius;
2. Prisijungti prie internetinės aplikacijos;
3. Registruotis prie internetinės aplikacijos.

Registruotas sistemos naudotojas galės:

1. Atsijungti nuo internetinės aplikacijos;
2. Užpildyti asmeninę susisiekiimo informaciją skelbimams;
3. Sukurti skelbimus:
 - 3.1. Įkelti automobilio ar dalies nuotrauką;
 - 3.2. Pridėti informaciją apie automobilį ar dalį;
 - 3.3. Pridėti aprašymą apie automobilį ar dalį;
4. Įdėti skelbimą į katalogą;

5. Peržiūrėti kitų registruotų naudotojų skelbimus;

Administratorius galės:

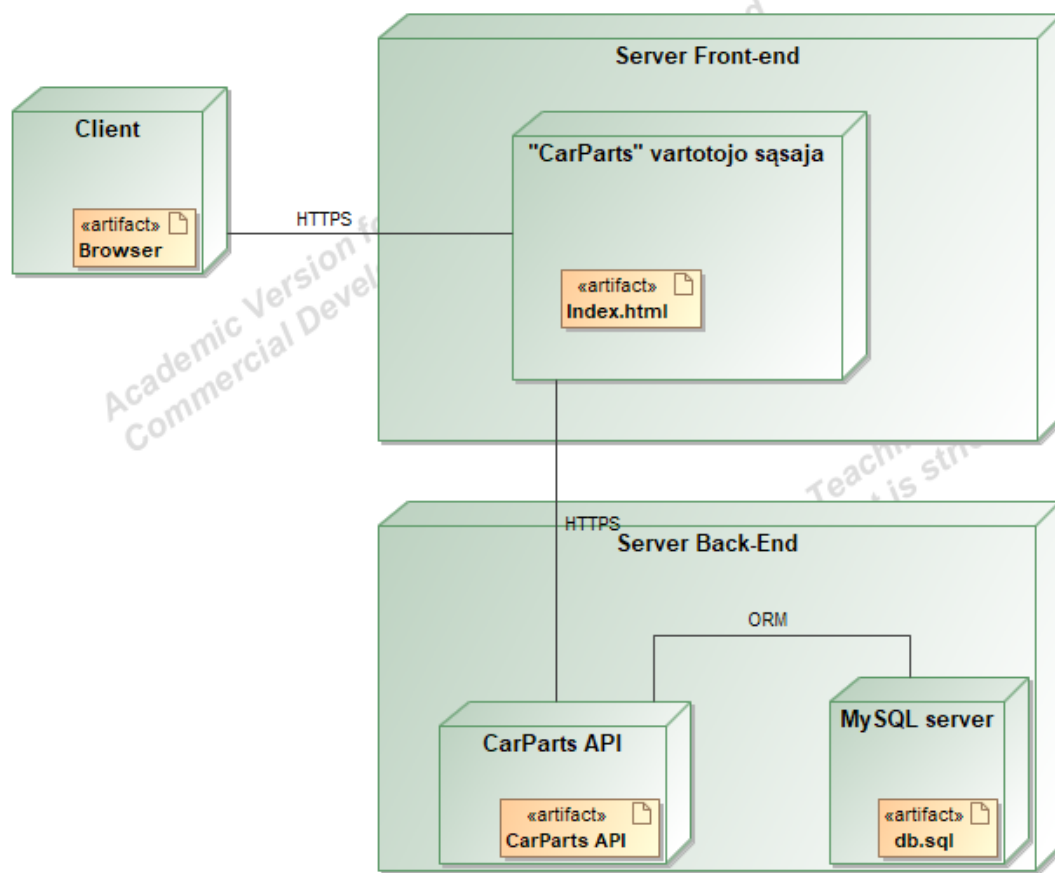
1. Kurti automobilių markes, automobilį, automobilio dalis kataloge;
2. Turi visas galimybes, kaip prisijungę vartotojai.

2. Sistemos architektūra

Sistemos sudedamosios dalys:

- Kliento pusė (ang. Front-End) – naudojant React.js;
- Serverio pusė (angl. Back-End) – naudojant C#, Visual Studio 2022, .Net Core API.
Duomenų bazė – Microsoft SQL

(2.1 pav.) pavaizduota kuriamos sistemos diegimo diagrama. Sistemos talpinimui yra naudojama lokali MySQL duomenų bazė. Internetinė aplikacija yra pasiekiamą per HTTP protokolą. Šios sistemos veikimui (pvz., duomenų manipuliavimui su duomenų baze) yra reikalingas Postman API, kuris pasiekiamas per aplikacijų programavimo sąsają. Pats Pharma API vykdo duomenų mainus su duomenų baze - tam naudojama ORM sąsaja.



2.1 pav. Sistemos „CarParts“ diegimo diagrama

3. API Specifikacija

Atsakymo formatas: JSON.

3.1. CarParts API metodai

Brands CRUD

Get All Brands	
API metodas	GET
Kelias iki metodo	/api/Brands
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Gauti visas automobilių markes
Užklauso "Header" dalis	-
Atsakymo struktūra	<pre>{ "id": 2, "name": "Audi", "description": "audi cars" },</pre>
Atsakymo kodas	200OK
Galimi klaidų kodai	404 – not found

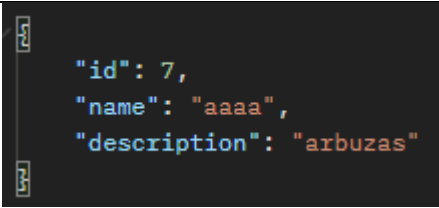
Gauto atsakymo pavyzdys	<pre>[{ "id": 2, "name": "Audi", "description": "audi cars" }, { "id": 3, "name": "Bmw", "description": "bmw cars" }, { "id": 4, "name": "Mercedes", "description": "Mercedes cars" }, { "id": 5, "name": "Volvo", "description": "Volvo cars" }, { "id": 6, "name": "Mitsubishi", "description": "Mitsubishi cars" }]</pre>
-------------------------	--

Get Brand	
API metodas	GET
Kelias iki metodo	/api/brands/{id}
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Gauti konkrečią markę
Atsakymo struktūra	<pre>[{ "id": 3, "name": "Bmw", "description": "bmw cars" }]</pre>
Atsakymo kodas	200OK
Galimi klaidų kodai	404 – not found

Gauto atsakymo pavyzdys	<pre> { "id": 2, "name": "Audi", "description": "audi cars" } </pre>
-------------------------	--

Create Brand	
API metodas	POST
Kelias iki metodo	/api/brand
Vartotojai, galintys pasiekti	Admin, User
Paskirtis	Sukurti ir pridėti naują markę į katalogą
Atsakymo struktūra	<pre> { "id": 7, "name": "name2", "description": "desc2" } </pre>
Atsakymo kodas	201 Created
Galimi klaidų kodai	401 - Unauthorized
Gauto atsakymo pavyzdys	<pre> { "id": 7, "name": "name2", "description": "desc2" } </pre>

Update Brand	
API metodas	PUT
Kelias iki metodo	/api/brand
Vartotojai, galintys pasiekti	Admin, User
Paskirtis	Atnaujinti markes
Atsakymo struktūra	<pre> { "id": 7, "name": "aaaa", "description": "arbuzas" } </pre>
Atsakymo kodas	200 OK

Galimi klaidų kodai	404 - Not found, 401 - Unauthorized
Gauto atsakymo pavyzdys	 <pre> {id": 7, "name": "aaaa", "description": "arbuzas" </pre>

Delete Brand	
API metodas	DEL
Kelias iki metodo	/api/brands
Vartotojai, galintys pasiekti	Admin, User
Paskirtis	Panaikinti markę
Atsakymo struktūra	-
Atsakymo kodas	204 – No content
Galimi klaidų kodai	204
Gauto atsakymo pavyzdys	-

Cars CRUD

Get All Cars	
API metodas	GET
Kelias iki metodo	/api/brands/{id}/cars
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Gauti visas mašinas
Atsakymo struktūra	<pre>{ "id": 1, "name": "Toyota Prius", "imageUrl": "https://cdn.vox-cdn.com/thumbor/Q-7lrklWyHIBdK15i2rlKoj2lcM=/258x59:3580x2160/3322x2101/filters:focal(1905x1382:1906x1383)/cdn.vox-cdn.com/uploads/chorus_asset/file/24202837/20221116_01_03.jpg", "engineSize": "2.0l", "fuelType": "Benzinas", "creationYear": "2019", "brandId": 2 }</pre>
Atsakymo kodas	200 - OK
Galimi klaidų kodai	200
Gauto atsakymo pavyzdys	<pre>{ "id": 1, "name": "Toyota Prius", "imageUrl": "https://cdn.vox-cdn.com/thumbor/Q-7lrklWyHIBdK15i2rlKoj2lcM=/258x59:3580x2160/3322x2101/filters:focal(1905x1382:1906x1383)/cdn.vox-cdn.com/uploads/chorus_asset/file/24202837/20221116_01_03.jpg", "engineSize": "2.0l", "fuelType": "Benzinas", "creationYear": "2019", "brandId": 2 }</pre>

Get Car	
API metodas	GET
Kelias iki metodo	/api/cars/{id}
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Gauti pagal automobilio id
Atsakymo struktūra	<pre>{ "id": 3, "name": "Bmw M3 e92", "imageUrl": "https://chrometemple.com/wp-content/uploads/2021/07/2008-BMW-E92-M3-MANUAL-SKU000007-PROFILE-scaled.jpg", "engineSize": "5.0l", "fuelType": "Benzinas", "creationYear": "2013", "brandId": 2 }</pre>
Atsakymo kodas	200 - OK
Galimi klaidų kodai	200, 404

Gauto atsakymo pavyzdys	<pre> { "id": 3, "name": "Bmw M3 e92", "imageUrl": "https://chrometemple.com/wp-content/uploads/2021/07/2008-BMW-E92-M3-MANUAL-SKU000007-PROFILE-scaled.jpg", "engineSize": "5.0l", "fuelType": "Benzinas", "creationYear": "2013", "brandId": 2 } </pre>
-------------------------	---

Get Car By Brand	
API metodas	GET
Kelias iki metodo	/api/brands/{id}/cars
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Gauti pagal markes
Atsakymo struktūra	<pre> { "id": 1, "name": "Toyota Prius", "imageUrl": "https://cdn.vox-cdn.com/thumbor/Q-7lrlWYHlBdK15i2rlKoj2lcM=/258x59:3580x2160/3322x2101/filters:focal(1905x1382:1906x1383)/cdn.vox-cdn.com/uploads/chorus_asset/file/24202837/20221116_01_03.jpg", "engineSize": "2.0l", "fuelType": "Benzinas", "creationYear": "2019", "brandId": 2 }, </pre>
Atsakymo kodas	200 - OK
Galimi klaidų kodai	400 – Bad Request
Gauto atsakymo pavyzdys	<pre> { "id": 1, "name": "Toyota Prius", "imageUrl": "https://cdn.vox-cdn.com/thumbor/Q-7lrlWYHlBdK15i2rlKoj2lcM=/258x59:3580x2160/3322x2101/filters:focal(1905x1382:1906x1383)/cdn.vox-cdn.com/uploads/chorus_asset/file/24202837/20221116_01_03.jpg", "engineSize": "2.0l", "fuelType": "Benzinas", "creationYear": "2019", "brandId": 2 }, </pre>

Create Car	
API metodas	POST
Kelias iki metodo	/api/brands/{id}/cars
Vartotojai, galintys pasiekti	Admin, User
Paskirtis	Sukurti automobilį

Atsakymo struktūra	<pre> { "Name" : "Toyota Prius", "EngineSize" : "2.0", "FuelType" : "Benzinas", "CreationYear" : "2019" } </pre>
Atsakymo kodas	201 Created
Galimi klaidų kodai	401 - Unauthorized
Gauto atsakymo pavyzdys	<pre> { "id": 8, "name": "Toyota Prius", "imageUrl": null, "engineSize": "2.0", "fuelType": "Benzinas", "creationYear": "2019", "brandId": 1 } </pre>

Update Car	
API metodas	PUT
Kelias iki metodo	/api/brands/{id}/cars/{id}
Vartotojai, galintys pasiekti	Admin, User
Paskirtis	Atnaujinti automobilį
Atsakymo struktūra	<pre> { "name": "Toyota Prius", "imageUrl": "https://cdn.vox-cdn.com/thumbor/Q-7lrkLWyHIBdK15i2rlKoj2lcM=/258x59:3580x2160/3322x2101/filters:focal(1905x1382:1906x1383)/cdn.vox-cdn.com/uploads/chorus_asset/file/24202837/20221116_01_03.jpg", "engineSize": "2.0", "fuelType": "electric", "creationYear": "2019" } </pre>
Atsakymo kodas	200 - OK
Galimi klaidų kodai	404 – not found
Gauto atsakymo pavyzdys	<pre> { "id": 4, "name": "Toyota Prius", "imageUrl": "https://cdn.vox-cdn.com/thumbor/Q-7lrkLWyHIBdK15i2rlKoj2lcM=/258x59:3580x2160/3322x2101/filters:focal(1905x1382:1906x1383)/cdn.vox-cdn.com/uploads/chorus_asset/file/24202837/20221116_01_03.jpg", "engineSize": "2.0", "fuelType": "electric", "creationYear": "2019", "brandId": 2 } </pre>

Remove Car	
API metodas	DEL
Kelias iki metodo	/api/brands/{id}/cars/{id}
Vartotojai, galintys pasiekti	Admin, User
Paskirtis	Panaikinti automobilį
Atsakymo struktūra	-
Atsakymo kodas	204 – No content
Galimi klaidų kodai	204, 404 – not found
Gauto atsakymo pavyzdys	-

Car Part CRUD

Get by Brand and Car	
API metodas	GET
Kelias iki metodo	/api/brands/{id}/cars/{id}
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Gauti automobilio dalis
Atsakymo struktūra	<pre> { "id": 1, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 }, { "id": 2, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 } </pre>
Atsakymo kodas	200 - OK
Galimi klaidų kodai	204, 404 – not found
Gauto atsakymo pavyzdys	<pre> { "id": 1, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 }, { "id": 2, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 } </pre>

Get All CarParts	
API metodas	GET
Kelias iki metodo	/api/carParts
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Gauti visas automobilio dalis
Atsakymo struktūra	<pre> { "id": 1, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 }, { "id": 2, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 } </pre>
Atsakymo kodas	200 - OK
Galimi klaidų kodai	404 – not found
Gauto atsakymo pavyzdys	<pre> { "id": 1, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 }, { "id": 2, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 } </pre>

Create CarPart	
----------------	--

API metodas	POST
Kelias iki metodo	/api/brands/{id}/cars/{id}/carParts
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Sukurti automobilio dalį
Atsakymo struktūra	<pre> { "Name": "test part", "ImageURL": "test URL", "Code": "test 123456789", "Description": "test test test" } </pre>
Atsakymo kodas	201 - Created
Galimi klaidų kodai	400 – Bad request
Gauto atsakymo pavyzdys	<pre> { "id": 2, "name": "test part", "imageURL": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 } </pre>

Update CarPart	
API metodas	POST
Kelias iki metodo	/api/brands/{id}/cars/{id}/carParts/{id}
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Atnaujinti automobilio dalį
Atsakymo struktūra	<pre> { "Name": "test part 1", "ImageURL": "test URL 1", "Code": "test 123456789 1", "Description": "test test test 1" } </pre>
Atsakymo kodas	200 - OK
Galimi klaidų kodai	400 – Bad request

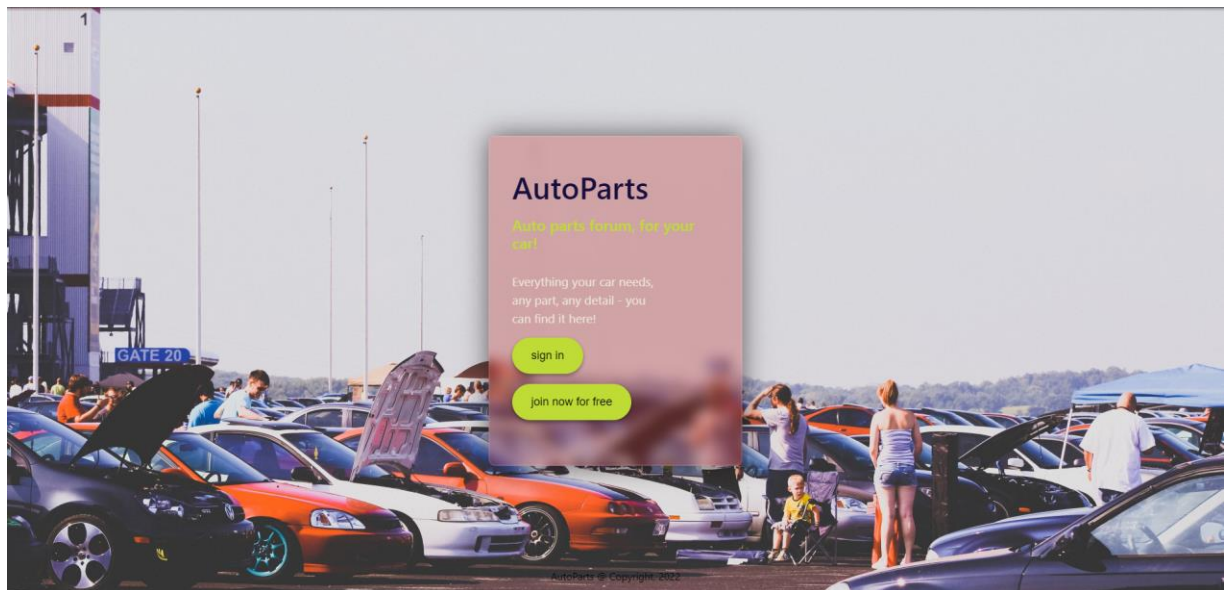
Gauto atsakymo pavyzdys	<pre> { "id": 2, "name": "test part", "imageUrl": "test URL", "code": "test 123456789", "description": "test test test", "carId": 3 } </pre>
-------------------------	--

Delete CarPart	
API metodas	DEL
Kelias iki metodo	/api/brands/{id}/cars/{id}/carParts/{id}
Vartotojai, galintys pasiekti	Admin, User, Guest
Paskirtis	Ištrinti automobilio dalį
Atsakymo struktūra	-
Atsakymo kodas	204 – No Content
Galimi klaidų kodai	400 – Bad request, 404 – not found
Gauto atsakymo pavyzdys	-

4. Svetainės Front-end įgyvendinimas

4.1. Main Menu langas

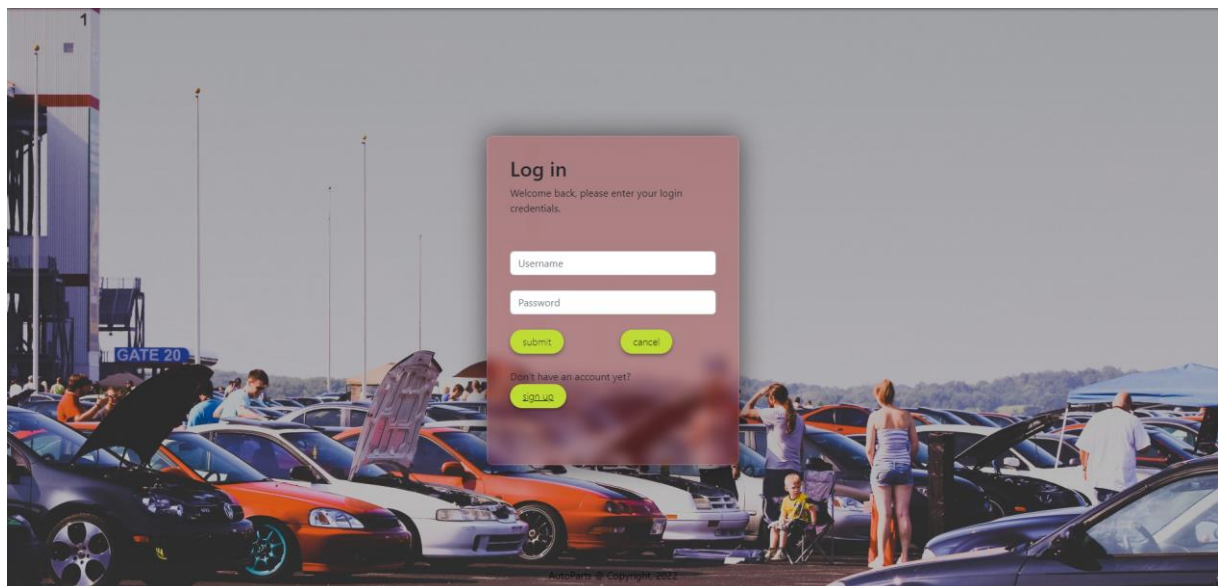
Api/ :



pav. 4.1 Main menu API/ langas

4.2. Login langas

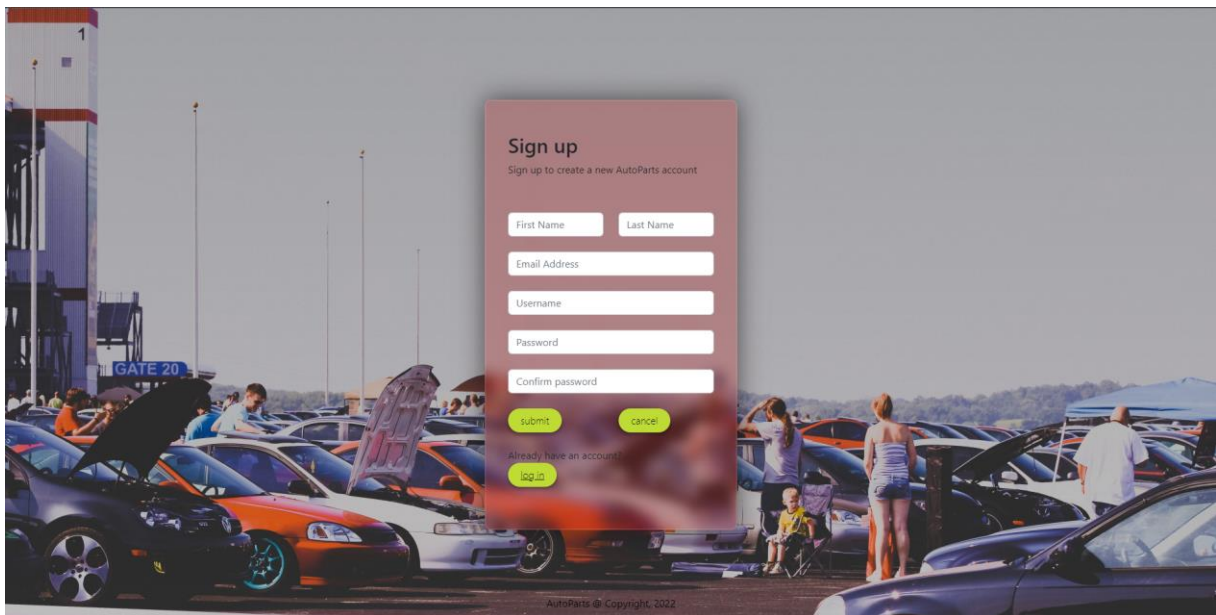
Api/Login :



4.2 pav. Login API/Login langas

4.3. Register langas

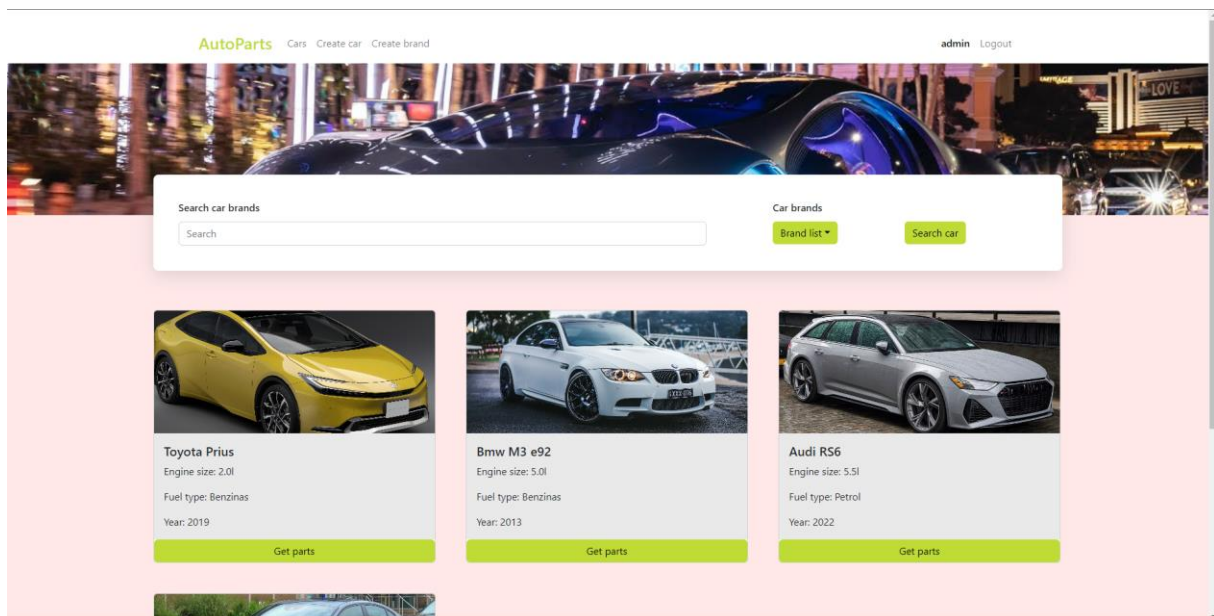
Api/Register :



4.3 pav. Register API/register langas

4.4. Automobilių langas

Api/cars :



4.4 pav. Cars API/cars langas

4.5. Automobilio kūrimo langas

Api/createCar :

AutoParts
Cars
Create car
Create brand
admin
Logout

Create new car

Car name

Image URL

Engine Size

Fuel Type

Creation Year

Brand

Create car

AutoParts © Copyright, 2022

4.5 pav. CreateCar API/createCar langas

4.6. Automobilio markės kūrimo langas

Api/createBrand :

AutoParts
Cars
Create car
Create brand
admin
Logout

Create new brand

Brand name

Description

Create brand

AutoParts © Copyright, 2022

4.6 pav. CreateBrand API/createBrand langas

5. Išvados

Saitynų taikomųjų programų projektavimo moduliui buvo sukurta svetainė, skirta žmonėms pagal jų automobilius susirasti savo dalis, jų informaciją ir gauti nuorodas į parduotuves, kurios turi tas dalis parduoti. Buvo suprogramuotas CRUD funkcijos su automobilių markėmis, automobiliais ir automobilių dalimis. Kuriant Back-end dalį buvo remtasi REST principais, įrankiais, kuriais buvo naudojama - .NET, Entity framework, duomenų bazė – SQL. Front-end dalis buvo sukurta pasinaudojant React biblioteka. Projekto pabaigoje sudaryta dokumentacija – aprašytas kiekvienas API kreipinys, kokius parametrus reikia paduoti bei koks yra numatomas rezultatas.