|  |  |  |
| --- | --- | --- |
|  | | |
| API tokeni | | |
|  |  |  |
| Kandidat |  | Mentor |
| Jevanđel Đurić |  | DR zlatko čović |
| 28124010 |  |  |
|  | Subotica, 2025 |  |

Sažetak

Projekat je unapređena verzija postojećeg projekta iz predmeta Napredno Web Programiranje. Dodata je mogućnost pristupa korisnika događajima iz baze podataka putem API tokena, mogućnost logovanja putem Google naloga i plaćanje putem Stripe platforme. Korisnički podaci u su smešteni u JWT token, koji se nakon logovanja korisnika verifikuje na svakoj stranici koju korisnik posećuje.

*API*

*Stripe*

*OAuth 2.0*

Sadržaj

[Sažetak 2](#_Toc983593553)

[Sadržaj 3](#_Toc2075214312)

[1. Rešavanje problema 4](#_Toc845030857)

[1.1. Struktura baze podataka 4](#_Toc1124255988)

[1.2. Stripe 4](#_Toc1809264010)

[1.3. Kreiranje tokena 5](#_Toc2079258197)

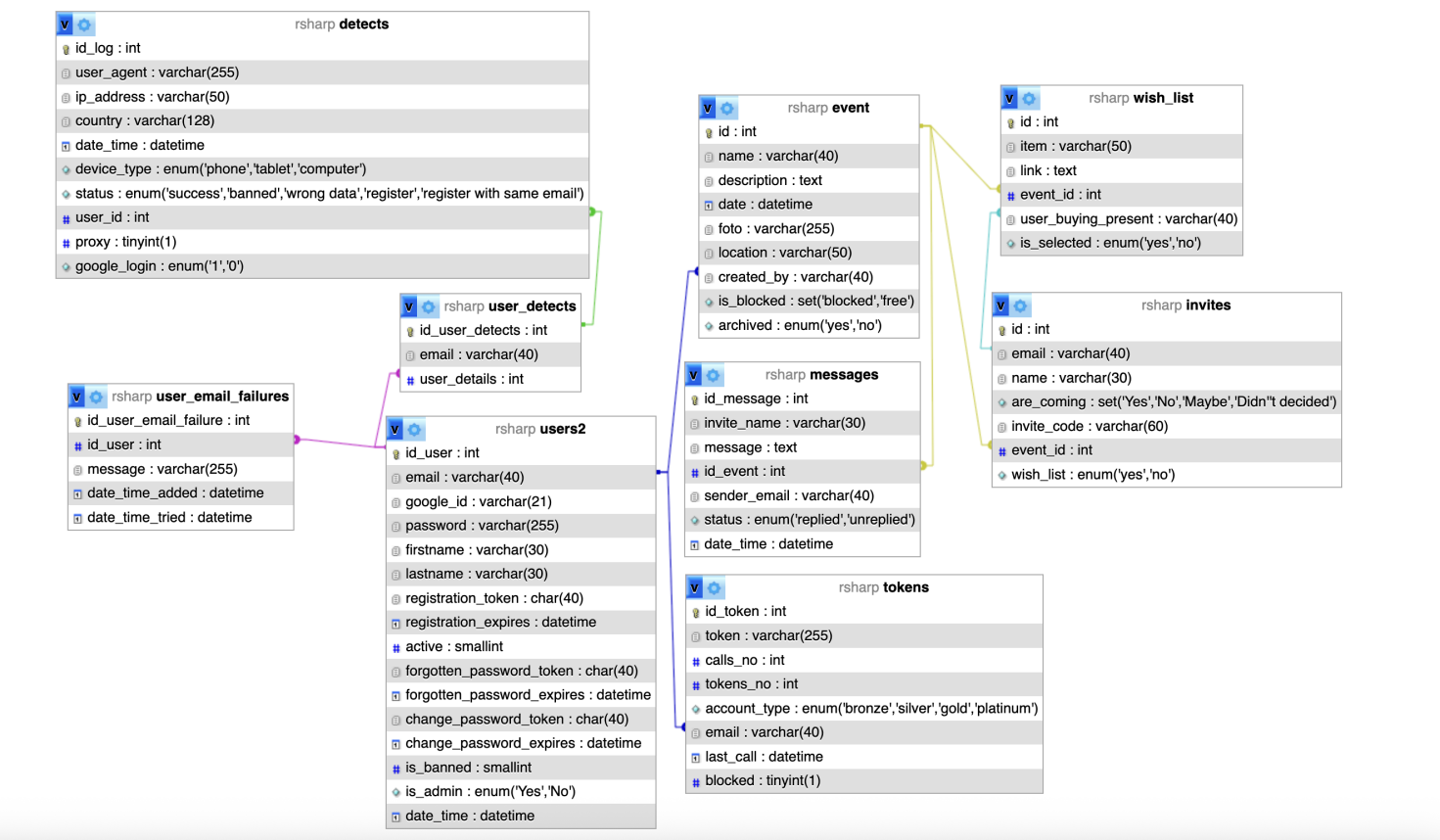
[1.4. JWT tokeni 6](#_Toc91129348)

[1.5. Google OAuth 2.0 login 7](#_Toc455111525)

[Literatura 8](#_Toc1870133708)

# Rešavanje problema

## Struktura baze podataka



## Stripe

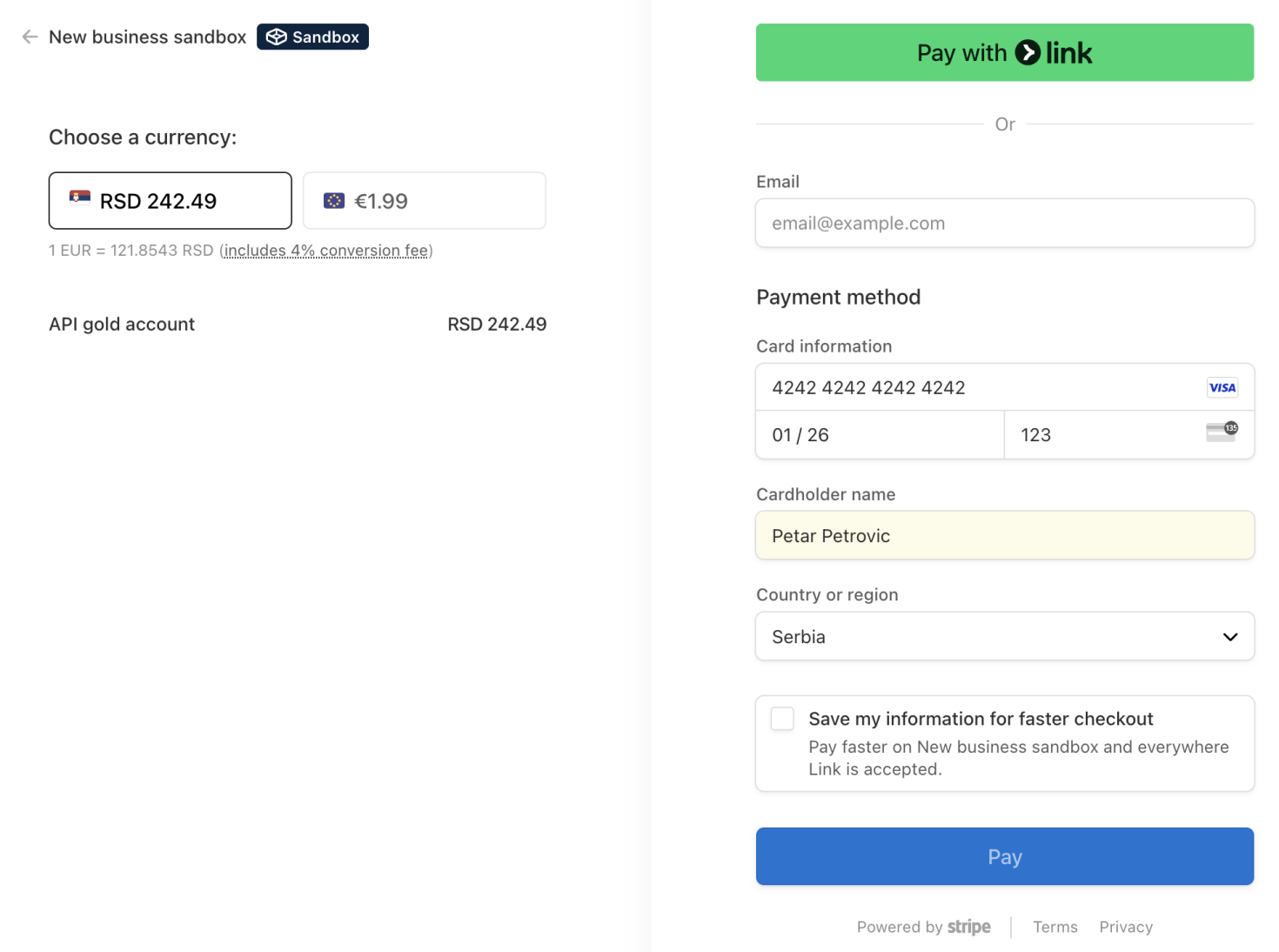
Stripe sam instalirao putem Composera

composer require stripe/stripe-php

Programski kod za podešavanje Stripe parametara plaćanja (checkout.php)

<?php  
require *\_\_DIR\_\_* . "/vendor/autoload.php";  
  
$dotenv = Dotenv\Dotenv::*createImmutable*(*\_\_DIR\_\_*);  
$dotenv->load();  
  
\Stripe\Stripe::*setApiKey*($\_ENV['stripe\_secret\_key']);  
  
$checkout\_session = \Stripe\Checkout\Session::*create*([  
 "mode" => "payment",  
 "success\_url" => "https://rsharp.stud.vts.su.ac.rs/successful\_payment.php?session\_id={CHECKOUT\_SESSION\_ID}",  
 "cancel\_url" => "https://rsharp.stud.vts.su.ac.rs/index.php",  
 "line\_items" => [  
 [  
 "quantity" => 1,  
 "price\_data" => [  
 "currency" => "eur",  
 "unit\_amount" => (int)$\_POST['price'],  
 "product\_data" => [  
 "name" => 'API '.$\_POST['plan'].' account'  
 ]  
 ]  
 ]  
 ],  
 "metadata" => [  
 "email" => $\_POST['email'],  
 "plan" => $\_POST['plan']  
 ]  
]);  
  
http\_response\_code(303); *//informacija o redirekciji sa mog sajta*header("Location: " . $checkout\_session->url);

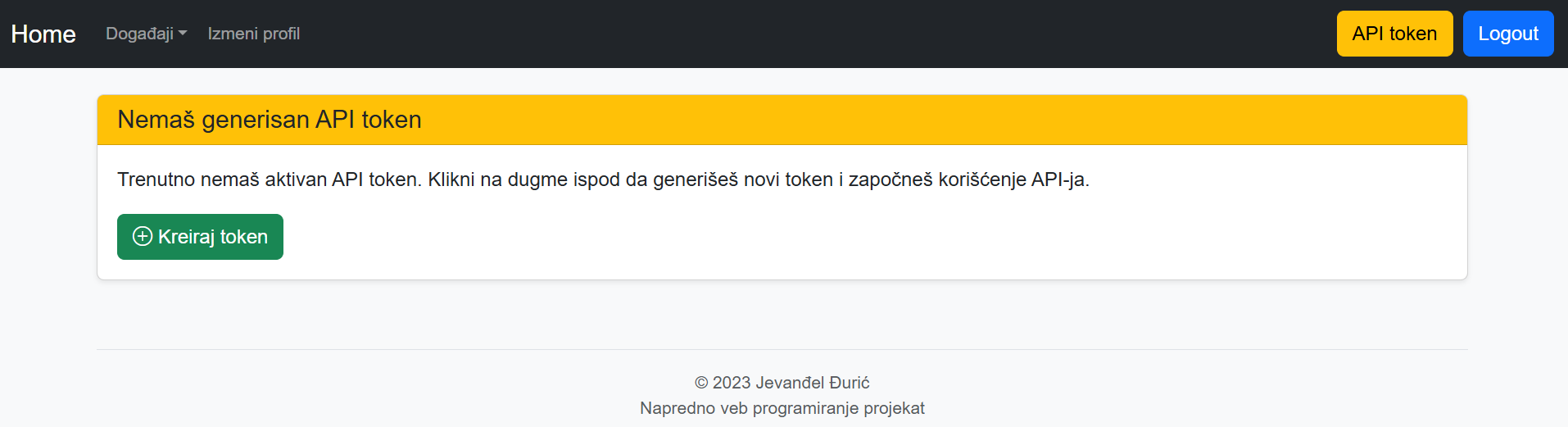
Za uspešno probno plaćanje moguće je koristiti VISA karticu broj: 4242 4242 4242 4242 (Slika 1). Kompletan primer je dat na <https://docs.stripe.com/checkout/quickstart#testing>



Slika 1. Probno plaćanje

## Kreiranje tokena

Za pristup API funkcionalnosti sajta potrebno je da korisnik na svom profilu generiše API Bearer token (Slika 2).



Slika 2. Kreiranje tokena preko sajta

Takođe moguće je generisati token putem API poziva na endpoint https://rsharp.stud.vts.su.ac.rs/api/tokens/ ukoliko korisnik ima nalog na sajtu i nema već kreiran token. U telu zahteva je potrebno navesti i JSON parametre u datom formatu:

{

“email”:”26121057@vts.su.ac.rs”

}

## JWT tokeni

Za zaštitu promenljivih sesije u koje se smeštaju korisnički podaci promenljive su enkodovane u JWT token na login.php/register.php stranici. Ključ za validaciju podataka je smešten u skrivenu .env datoteku. Da bi pojednostavio kreiranje i verifikaciju tokenda dodao sam composer biblioteku:

composer require firebase/php-jwt

Primer enkodovanja korisničkih podataka prilikom logovanja/registracije:

$dotenv = Dotenv\Dotenv::*createImmutable*(*\_\_DIR\_\_*); *//Uključivanje ENV fajla*  
$dotenv->load();

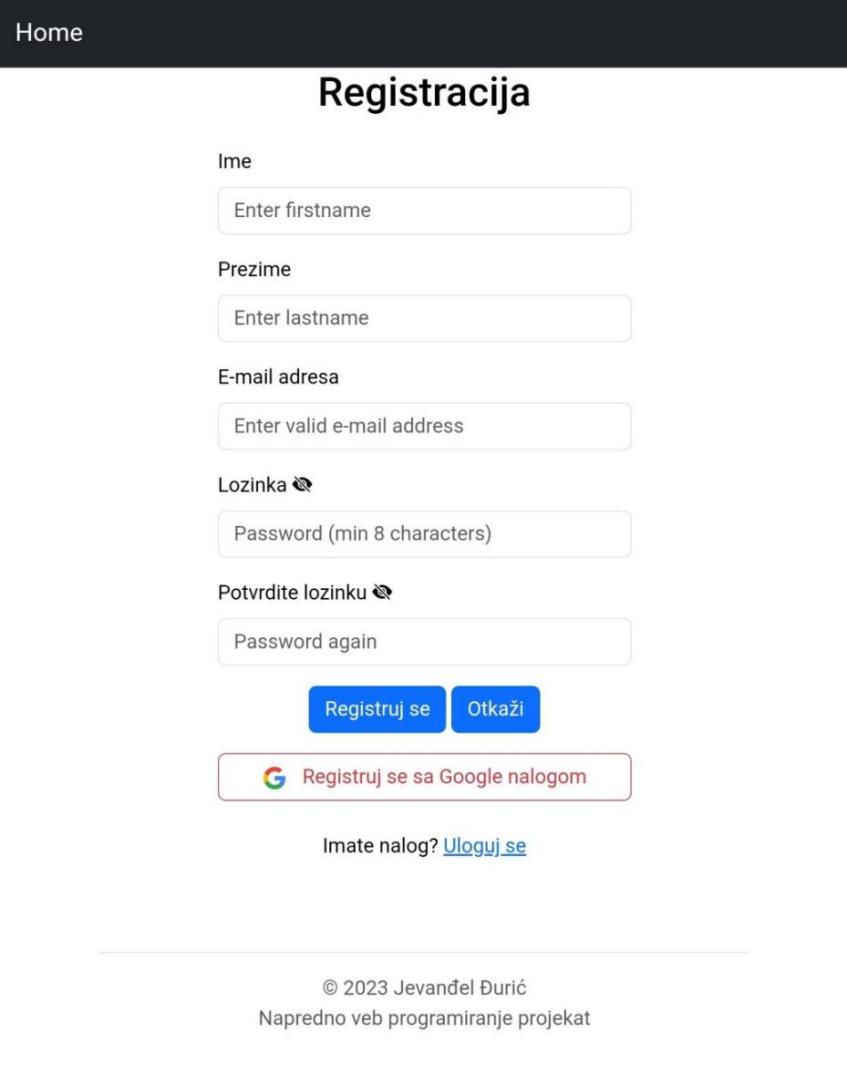
$payload=[  
 'iss' => 'https://rsharp.stud.vts.su.ac.rs',  
 'aud' => 'https://rsharp.stud.vts.su.ac.rs',  
 'iat' => time(),  
 'nbf' => time(),  
 'exp' => time()+(60\*60),  
 'data' => [  
 'username'=>$userInfo['email'],  
 'firstname'=>$data['firstname'],  
 'is\_admin'=>$data['is\_admin'],  
 'id\_user'=>$userData  
 ]  
];  
$encoded = JWT::*encode*($payload, $\_ENV['jwt\_secret\_key'], 'HS256');  
$\_SESSION = [];  
$\_SESSION['jwt\_token'] = $encoded;

Dekodovanje na korisničkim stranicama:

if (isset($\_SESSION['jwt\_token'])) {  
 try {  
 $decoded = JWT::*decode*($\_SESSION['jwt\_token'], new Key($\_ENV['jwt\_secret\_key'], 'HS256'));  
 }catch (\Firebase\JWT\ExpiredException){  
 redirection('login.php'); *//Ukoliko je token istekao tražiti ponovno logovanje*  
 }  
}

## Google OAuth 2.0 login

Sajtu je dodata mogućnost kreiranja naloga putem google naloga



Potrebno je da korisnik prvo kreira nalog na sajtu klikom na dugme “Registruj se sa Google nalogm”. Ukoliko korisnik pokuša da se uloguje se google nalogom ukoliko se prethodno nije registrovao biće mu prikazano odgovarajuće obaveštenje.

Dodata je composer biblioteka

composer require google/apiclient

Za dobijanje parametara bilo je potrebno kreirati google cloud projekat na sajtu <https://console.cloud.google.com>

$client = new Google\Client();  
  
$client->setClientId("912256885401-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*.apps.googleusercontent.com");  
$client->setClientSecret("GOCSPX-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
$client->setRedirectUri("https://rsharp.stud.vts.su.ac.rs/redirectLogin.php");  
  
$client->addScope("email");  
$client->addScope("profile");  
  
$url = $client->createAuthUrl();

Literatura

1. <https://docs.stripe.com/>
2. <https://getbootstrap.com/docs/5.3>
3. <https://developers.google.com/identity/protocols/oauth2>
4. <https://stackoverflow.com/>
5. <https://github.com/Andjelo12/BUEP-projekat>
6. <https://chat.openai.com>