**Security Documentation and Implementation**

**GitHub link- <https://github.com/thisalma/laravel_project.git>**

This Document describes the security measures implemented throughout the Foodies laravel application to protect both web and mobile users.

1. **User Authentication- laravel Jetstream**

The web application uses Laravel Jetstream and Fortify for user authentication. Routes are protected to ensure that only authenticated users can access certain pages. Email verification is enforced for added security.  
  
Code:-

@if (Laravel\Fortify\Features::enabled(Laravel\Fortify\Features::emailVerification()) && ! $this->user->hasVerifiedEmail())

                <p class="text-sm mt-2">

                    {{ \_\_('Your email address is unverified.') }}

                    <button type="button" class="underline text-sm text-gray-600 hover:text-gray-900 rounded-md focus:outline-none focus:ring-2 focus:ring-offset-2 focus:ring-indigo-500" wire:click.prevent="sendEmailVerification">

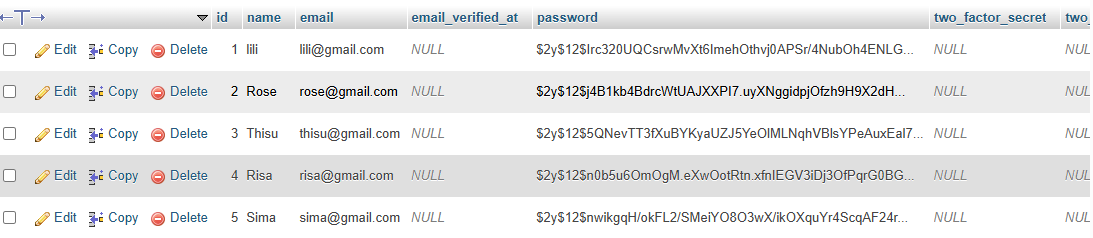
                        {{ \_\_('Click here to re-send the verification email.') }}

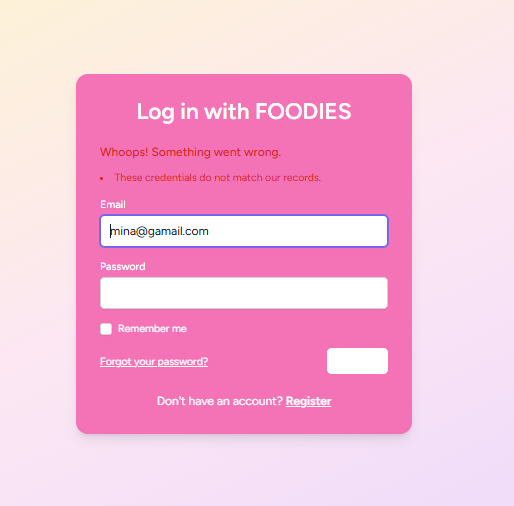
                    </button>

                </p>

Explanation:- Ensures only verified emails can access Jetstream handles authentication and protects routes.

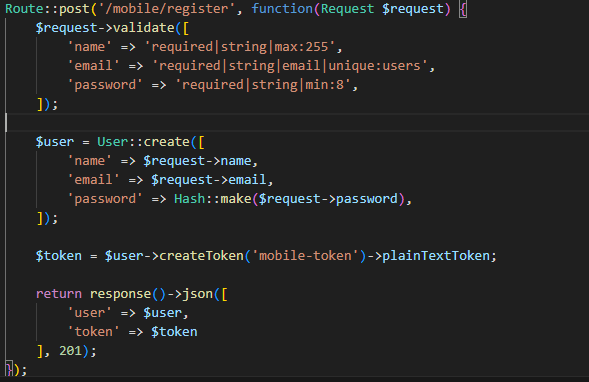
* In the database the once who has being registered details are being stored

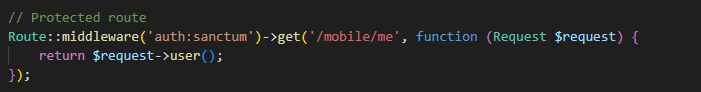


* When we try to login with a data which is not registered and store on the database an error will be shown like this  
  

1. **Authentication (API) - Laravel Sanctum**

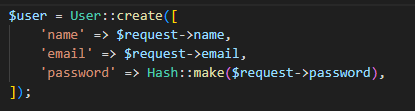
The mobile API is secured using Laravel Sanctum, providing token based authentication for API endpoints. Only authenticated users with valid tokens can access protected routes.





1. **Password Security**

* Passwords are hashed using Laravel Hash before storing in the database.
* Password rules enforce minimum 8 characters.

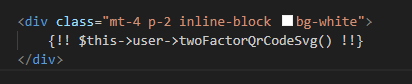




Prevents plain text password storage, reducing risk if the database is compromised.Even if stolen, hashed passwords cannot be used directly.

1. **Two Factor Authentication**

Implemented using: Jetstream’s TwoFactor Authentication Livewire component.



Security Benefits:

Adds extra layer of protection against stolen passwords.

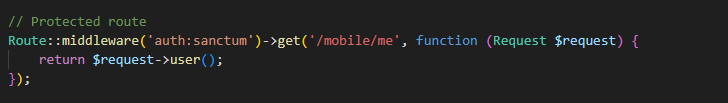
Recovery codes ensure access recovery in case device is lost.

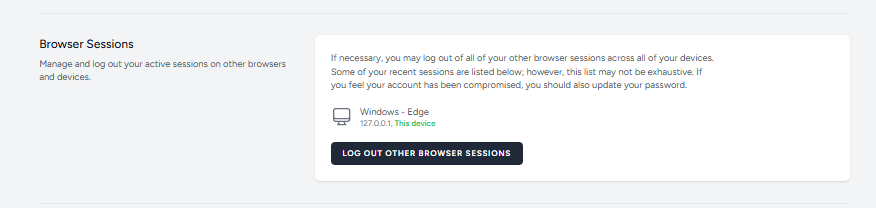
1. **Session Management**

Implemented using: Livewire components for managing browser sessions.

Features:

* Users can view active sessions and log out of other devices.
* Each session stores IP, browser, and last activity.





Security Benefits:

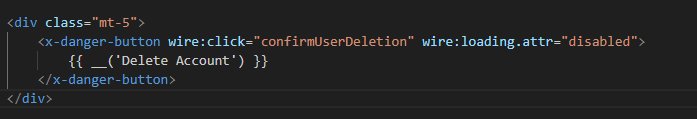
* Prevents unauthorized access if session is hijacked.
* Allows user to terminate all other active sessions.

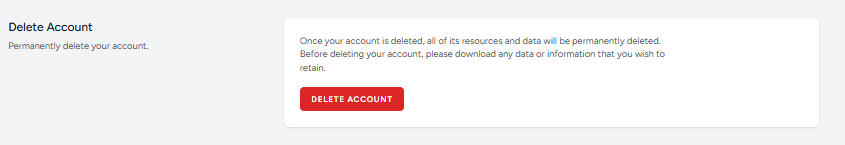
1. **Account Deletion & Confirmation**

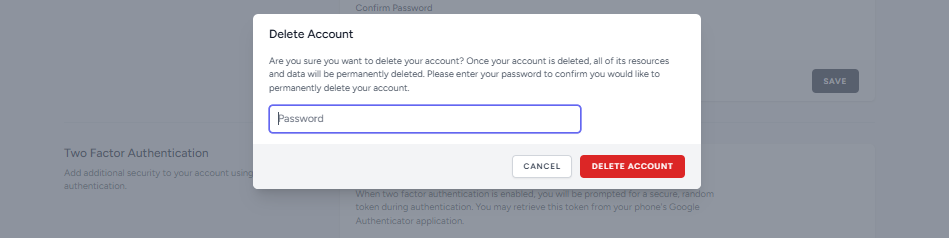
Implemented using: delete-user-form Livewire component.

Features:

* User must confirm password before deletion.
* All user data and resources are permanently deleted.







Security Benefits:

Prevents accidental or unauthorized account deletion.

1. **API Security (Mobile App)**

Implemented using: Laravel Sanctum.



Features:

* API tokens are issued during mobile registration and login.
* Protected routes require auth:sanctum middleware.

**Security Test case**

| **Security Feature** | **Test Case** | **Expected Result** | **Status** |
| --- | --- | --- | --- |
| User Registration | Register with valid email & password | Account created, password hashed | Pass |
| Duplicate Registration | Register with existing email | Validation error | Pass |
| User Login | Login with correct credentials | Access granted, API token generated | Pass |
| Invalid Login | Login with wrong credentials | Login denied, error message | Pass |
| Password Hashing | Check database for stored password | Password should be hashed | Pass |
| Two-Factor Authentication (2FA) | Enable 2FA and scan QR code | OTP required during login | Pass |
| 2FA Recovery Codes | View recovery codes | Codes displayed and can be used | Pass |
| Session Management | Logout from other sessions | Other sessions terminated | Pass |
| Account Deletion Confirmation | Delete account with/without password confirmation | Account deleted only after password confirmation | Pass |
| API Authentication (Sanctum) | Access /mobile/me without token | Access denied | Pass |
| API Token Access | Access /mobile/me with valid token | Access granted | Pass |
| Password Update | Update password with correct current password | Password updated, new login works | Pass |
| NoSQL Database (Optional) | Use MongoDB for API | N/A (SQL used) | NA |