# Navigation:

## Home page

* Navigation

Guests:

* + Login
  + Register
  + Rss Feed (as icon in the nav bar)

Logged-in Users:

* + Logout (under profile name dropdown in the nav bar)
  + User Articles (as 'My Articles')

All Users:

* + Home (as 'NewsStand')
  + Rss Feed (as icon in the nav bar)
* List of latest articles, each article title being a link to the article

## Register

(redirects to)

* Verificaion Notice (an email stub page with the verification link overrides the email notification for convenience for the demo)

## Email Stub

Redirects to (temporarily logged in):

* Activation form (setting of password required)

## Activation Form

Redirects to (logged in):

* User Articles

## Login

Redirects to:

* User Articles

**User Articles**

Has navigational links to:

* Navigation
* Articles (by clicking on their titles in the list)
* New Article (as 'New Article' button)
* Delete article in the list
* Download article

## New Article Form

Redirects to (after creation):

* Navigation
* User Articles

## Article

Redirects to (after creation):

* Navigation
* Download article
* Add Photo (when current user is the author)
* Delete article (when current user is the author)

## Add Photo Form

Redirects to (after selecting and uploading a file):

* Navigation
* Add Photo Form (to itself, for further image uploads)
* Article (go back)

# Code Organization and Component Interaction

## Routes

* Registered in routes/web.php, receiving the default middleware for the web route group.
* Designed to be stateless (apart from the user being authenticated) and RESTful, providing necessary parameters in the URI.
* Named and associated with controller methods for better convenience of use, extensibility and testability.
* Registered a regex pattern validation for id parameters in URIs in the app/Providers/RouteServiceProvider.class

## Controllers

* Found in app/Http/Controllers
* Authentication:
* Modified Auth/RegisterController to exclude password, as by requirement it is to be provided after email verification.
* Modified Auth/VerificationController to provide the email stub for convenience of demo presentation.
* Added Auth/ActivationController for setting the password and activating the user account.
* ArticleController:
* Manages requests from the routes related to articles.
* PhotoController:
* Manages requests from the routes related to photo uploads.
* MakesVerification trait:
* Included int article and photo controllers – provides methods for checking if an article exists and the current user is the author of the article.

## Middleware

* Found in app/Http/Middleware
* EnsureUserIsActive – verifies that the user has provided a password after email verification (registered in the app/Http/Kernel.php files in the $routeMiddleware and its priority set in $middlewarePriority)

## Commands

* An Artisan console command is defined in app/Console/commands/GenerateFeed and registered in app/Console/Kernel.php, which generates a fresh copy of the rss.xml file.

## Events

* Found in app/Events
* ArticleDeleted – fired by the ArticleRepository::delete() method.
* ArticleCreated – fired by the ArticleRepository::write() method.

## Listeners

* Found in app/Listeners
* DeleteNestedPhotos – deletes the photo models and image files associated with an article. Called when an ArticleDeleted event is fired.
* UpdateRssFeed – runs a console command to replace the rss.xml with a fresh one. Called when ArticleDeleted or ArticleCreated events are fired.
* SendEmailVerificationNotification – fired by the Registered event, when an email has been provided.

## Models

* Found in app/Models and app/ (the User model's default location)
* User
* overriden sendEmailVerificationNotification method so as to notify via an email stub, instead of actual email being sent.
* Added relationship to articles
* Add trait checking if account has been activated
* Article
* Photo

## Notifications

* Found in app/Notifications
* VerifyEmailDatabaseStub – a database notification, recording the verification link address

## Providers

* Bootstrapping classes found in app/providers
* AppServiceProvider – registers the interface-contrete class mappings and singletons in the IoC container.

## Repositories

* found in app/repositories
* EloquentRepository – implements basic RepositoryInterface functionality for Eloquent models.
* ArticleRepository – implements the article specific functionalities – extends EloquentRepository and implements ArticleRepositoryInterface.

## Summerizers

* Classes to be used as different strategies for making brief versions of the articles by the Article::preview() method which accepts Summerizing interface instances.
* NewsStandSummerizer – used for the article list in the NewsStand page.
* RssSummerizer – used for the article list in the rss feed.

## Configuration

* App-specific configuration found in config/newsportal.php

## Database

* Fond in the default Laravel location - /database folder, factories migrations and seeds are provided
* Separate migration files for the users table modification and foreign key constraints for more flexibility.
* Two users are created with emails [user1@mail.com](mailto:user1@mail.com) and [user2@email.com](mailto:user2@email.com) and passwords '123'.

## Resources

* Blade view templates for generating the html responses in the cotrollers, are found in the default location - /resources.views
* Translation strings are used for labels to allow for future internationalization (when missing as keys from the translation files, the keys themselves are used).
* Uploaded images are saved in public/img.
* Rss.xml is generated in storage\app\public.

## Tests

* Found in tests/Unit.
* ArticlesRepositoryTest