ASTON ANIMAL SANCTUARY REPORT

CS2410

INTERNET APPLICATIONS AND TECHNIQUES

CS2410

Appiah, Emmanuel (Student)

190079330

Emmanuel Appiah – 190079330
Candidate no – 965064
CS2410 – Internet Applications and Techniques

1 Details:

System Link: 190079330aas.freecluster.eu or alternative http://190079330.cs2410-web01pvm.aston.ac.uk

Login Details:

Public User Email: public@public.com Password: 12345678

Staff Email: admin@admin.com Password: 12345678

Administrator Email: versatile@user.com Password: 12345678

2 Brief Description of the Structure and Technologies

My Website, Aston Animal Sanctuary was created using Laravel which is seen as the current best PHP framework in existence. (M, 2021). It was combined with other tools such as SQL on apache server and other frontend frameworks such as Bootstrap and CSS. I started by creating a simple custom navigation bar and a homepage that links every page in my system. The system is developed for users to be able to adopt and manage their adoptions of animals. Registration and Login forms were also created to enable new users to carry about activities within the system. A variety of security features were also implemented in the system, which includes but are not limited to form validations, hash passwords, cross-site request forgery. These features which are mostly inbuilt in Laravel were effectively utilised, limiting the access to certain pages by public users and staff users.

Every user's first access to the website is mostly on the homepage, which he is greeted and given a general Welcome to the system. This page mainly consists of 4 pages (Home, Animals, Admin and About US).

Animals Page: On this page, any user regardless of its status can view the kinds of animals that are on the system and their basic information.

Admin Page: This page consists of a dropdown where a user can choose whether to login or to sign up.

About US Page: This page Consists of the general information about the Aston Animal Sanctuary and what it does.

All users basically can have access to certain general functions in the system, when logged in. For example, both public and staff users can logout and view the list of animals being given for adoption. Admin users have the same level of access as a logged in user as well as additional access to the adoption and user management pages. The admin can see all adoptions, their details, and can approve and reject them as they wish.

3 Model-View-Controller Implementation:

The Laravel framework allowed for creation of this system, based on the Model view controller Software Architecture. This architecture allowed for the structuring of the system directories in an organised manner. For example, model, view and controllers can be found in App/Http/Models, Resources/Views/ and App/Http/Controllers respectively. All these files are linked with a navigation source from the routes/web.php file. A database file (env) is also located withing the root directory of the system to allow creation of the details of the system.

Overall, the public directory in the root folder controls the central access of the website when it is hosted on the server.

4 Required Functions

Function	Properties	Location
User CRUD Functions	These are group of functions that allows users of the system to create, update, show, edit and delete items from the database in the system	UsersController.php
Animal CRUD Functions	These include all necessary functions that allows system users to carry out basic animal data functions and so on.	AnimalController.php
Adoption CRUD functions	These are the function that manages the view, update, delete and other manipulation functions of the adoption View.	AdoptionsController.php
Adoption Relation Functions • users() • animal()	These functions establish relations amongst users and animals.	Models/Adoptions.php
Animal Relation functions: • adoptions()	These functions establish relations between animals and adoptions	Models/Animal.php
Role Relation Functions: • User()	These functions establish relations between role the user models.	Models./Role.php
User Relation Functions: • Roles() • hasAnyRoles($roles) • hasRole($role) • adoptions()	These functions are created within the user model to establish relationships amongst roles and adoptions.	Models./User.php
Gates authentication Functions: • Gate::define('manage-users', function($user))	These functions are called in various view files to allow certain users to have permission over certain functions in the system.	Http/providers/authServiceprovider.php

• Gate::define('create-animal-record', function($user)) • Gate::define('edit-users', function($user)) • Gate::define('delete-users', function($user)) • Gate::define('approve-adoption-request', function($user)) • Gate::define('make-adoption, function($user)) • Gate::define('view-adoption-request, function($user))		

5 Security Features

Security Feature	Properties	File (s)
Form Validation	Ensure all fields are entered correctly. This includes checking the user has entered their email date or time in correct format	• Admin/edit.blade.php • Animals/create.blade.php • Animals/edit.blade.php • adoption/edit.blade.php • adoption/create.blade.php • auth/login.blade.php • auth/register.blade.php • auth/verify.blade.php • auth/passwords/confirm.blade.php • auth/passwords/email.blade.php • auth/passwords/reset.blade.php
Laravel Cross-site request forgery (XSS Cross-site Script)	Takes data user is inputting or viewing and changes special characters into html brackets and quotes into entities. The bowser renders them safely.	• Admin/edit.blade.php • Animals/create.blade.php • Animals/edit.blade.php • adoption/edit.blade.php • adoption/create.blade.php • auth/login.blade.php • auth/register.blade.php • auth/verify.blade.php • auth/passwords/confirm.blade.php • auth/passwords/email.blade.php • auth/passwords/reset.blade.php
Hash Passwords	Encrypt the password before sending and receiving it from the database. My system hashes passwords through the use of the MD5 method.	• auth/login.blade.php • auth/register.blade.php • auth/verify.blade.php • auth/passwords/confirm.blade.php • auth/passwords/email.blade.php

		- auth/passwords/reset.blade.php
Gates permission Authentication	Creates a situation, where different kinds of users have access to various kinds of pages.	- Http/providers/authServiceprovider.php

6 Database schema

My database schema consisted of the users, animals, adoptions and Roles table.

Users Table	Animals Table	Adoptions Table	Roles Table
Bigincreaments(id) PK	Bigincreaments(id) PK	Bigincreaments(id) PK	Bigincreaments(id) PK
String(‘name’)	Name(‘name’,20)->nullable	bigInteger('user_id')-> unsigned (FK)	String('name')
String (‘email’)->unique	Date(‘dateofbirth’)->nullable	bigInteger('animal_id')->unsigned (FK)	
Timestamp(‘email_verified_at’)->nullable	String(‘description’,256)->nullable	string('reason',256)->nullable	
String(‘password’)	Enum(‘Availability’,[‘Yes’,’No’])->default(‘Yes’);	enum('status', ['processing','approved','rejected'])->default('processing')	
rememberToken			

7 Implemented Stretchers

Stretchers	Details
User Management with Laravel Gates	Allows one versatile user, who can be a senior staff to edit other users permission.

8 References.

- M, W., 2021. *8 Best PHP Frameworks for Web Developers*. [online] Hostinger Tutorials. Available at: <https://www.hostinger.com/tutorials/best-php-framework#:~:text=1%20Laravel.%20When%20people%20talk%20about%20PHP%20frameworks%2C,CakePHP.%20...%207%20Yii.%20...%208%20FuelPHP.%20> [Accessed 29 April 2021].