

## **Internet Applications and Techniques Coursework Report**

### **Basic Information about development**

- Name and Student ID: Miraj Shah, 160050110
- Link to webpage: <https://www.cs2410-web02pvm.aston.ac.uk/~shahm23/>
- GitHub (link if needed): <https://github.com/miraj26/AstonAnimalSanctuary>
- Staff login details: username: admin                      password: admin
- User login details: username: bobby19                      password: password

### **Brief description of key technologies and structure of your system**

To complete this project, I used the Laravel framework and adhered to the MVC structure, where the views can be found in the resources folder, the controller and models can be found in the app folder. The database was created using phpMyAdmin using MySQL.

### **Required functions implemented**

Feature	Source File(s)
Register new user account	<ul style="list-style-type: none"> <li>• Auth\RegisterController.php</li> <li>• Auth\register.blade.php</li> </ul>
Login with user showing available pets	<ul style="list-style-type: none"> <li>• LoginController.php (not in the auth folder)</li> <li>• Auth\login.blade.php</li> <li>• HomeController.php → pets()</li> <li>• adoption_requests\availablepets.blade.php</li> </ul>
User makes an adoption request	<ul style="list-style-type: none"> <li>• HomeController.php → requested()</li> <li>• adoption_requests\requestsmade.blade.php</li> <li>• adoption_requests\edit.blade.php</li> <li>• AdoptionRequestController.php</li> </ul>
View all requests made by user	<ul style="list-style-type: none"> <li>• adoption_requests\requestsmade.blade.php</li> <li>• HomeController.php → allrequest()</li> </ul>
Admin login	<ul style="list-style-type: none"> <li>• LoginController.php (not in the auth folder)</li> <li>• Auth\login.blade.php</li> <li>• DashboardController → pending()</li> <li>• pending.blade.php</li> </ul>
Approve/Deny adoption request	<ul style="list-style-type: none"> <li>• DashboardController.php → review()</li> </ul>
List all animals in system and show who owns them	<ul style="list-style-type: none"> <li>• AnimalController.php → index()</li> <li>• animals\index.blade.php</li> <li>• animals\owner.blade.php</li> <li>• UserController.php</li> </ul>
All Requests made by all users	<ul style="list-style-type: none"> <li>• DashboardController.php → requests()</li> <li>• admin\requests.blade.php</li> </ul>
Add new animal to system	<ul style="list-style-type: none"> <li>• AnimalController.php → create() and store()</li> </ul>

### **Security features used**

Feature	Source File(s)
---------	----------------

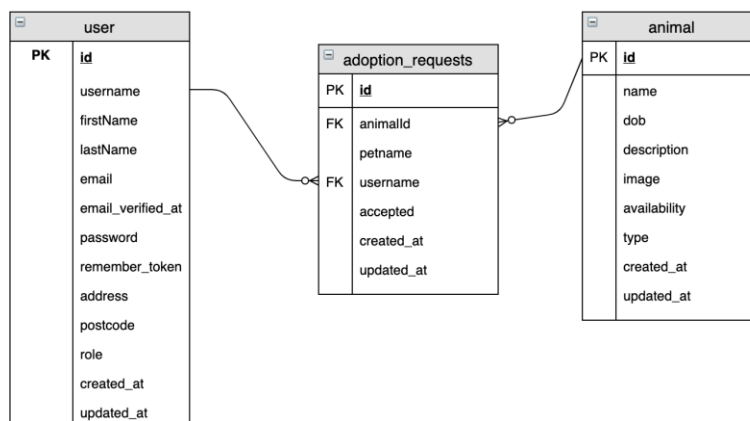
Authentication/Authorisation	<ul style="list-style-type: none"> <li>• Auth\VerificationController.php</li> <li>• web.php → group middleware for user and admin</li> <li>• LoginController.php → login()</li> </ul>
Form Validation	<ul style="list-style-type: none"> <li>• register.blade.php</li> <li>• Auth\RegisterController.php → validator()</li> <li>• login.blade.php</li> <li>• animals\create.blade.php</li> <li>• AnimalController.php → store() and update()</li> <li>• animals\edit.blade.php</li> <li>• edituser.blade.php</li> <li>• HomeController.php → edit() and update()</li> </ul>
Hash password	<ul style="list-style-type: none"> <li>• Auth\RegisterController.php → create()</li> </ul>
Restrict file upload to only images	<ul style="list-style-type: none"> <li>• AnimalController.php → store() and update()</li> </ul>
CSRF	<ul style="list-style-type: none"> <li>• Auth\register.blade.php</li> <li>• Auth\login.blade.php</li> <li>• edituser.blade.php</li> <li>• adoption_requests\edit.blade.php</li> <li>• animals\create.blade.php</li> <li>• animals\edit.blade.php</li> <li>• admin\pending.blade.php</li> </ul>
Restrict URLs- so you have to be logged in to view them (middleware)	<ul style="list-style-type: none"> <li>• web.php → group middleware for user and admin; each group contains a list of URLs, so they have to be logged into the write account in order to view them otherwise they will be redirected to the log in page</li> <li>• Middleware\Admin.php</li> <li>• Middleware\User.php</li> <li>• User.php → is_admin()</li> </ul>
Handling Injections	The Laravel framework used the Eloquent ORM, which uses PDO parameter binding to protect against SQL injections. Blade statements are automatically send through PHP's htmlspecialchars function to prevent again HTML injections.

### Stretchers

Feature	Source File(s)
Different animal types	<ul style="list-style-type: none"> <li>• migrations\ 2019_04_03_194234_create_animals_table.php</li> <li>• AnimalController.php → store()</li> </ul>
Filter Animals	<ul style="list-style-type: none"> <li>• AnimalController.php → index()</li> <li>• animals\index.blade.php</li> <li>• HomeController.php → pets()</li> <li>• adoption_requests\availablepets.blade.php</li> </ul>

User can view their own information	<ul style="list-style-type: none"> <li>• HomeController.php → index()</li> <li>• home.blade.php</li> </ul>
View the details of the owner through the admin view	<ul style="list-style-type: none"> <li>• animals\owner.blade.php</li> <li>• UserController.php → show()</li> </ul>
User can edit some of their details	<ul style="list-style-type: none"> <li>• edituser.blade.php</li> <li>• HomeController.php → update() and edit()</li> </ul>
Pagination	<ul style="list-style-type: none"> <li>• AnimalController.php → index()</li> <li>• HomeController.php → pets()</li> </ul>
Admin can view, edit and delete a pet	<ul style="list-style-type: none"> <li>• AnimalController.php: display → show(); edit → edit() and update(); delete → destroy;</li> </ul>

### Database Schema



- A user can make multiple requests to different animals and an animal can have many requests from different users.

### Users Table Constraints:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b>	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>username</b>	varchar(50)	utf8mb4_unicode_ci		No	None		
3	<b>firstName</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
4	<b>lastName</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
5	<b>email</b>	varchar(100)	utf8mb4_unicode_ci		No	None		
6	<b>email_verified_at</b>	timestamp			Yes	NULL		
7	<b>password</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
8	<b>remember_token</b>	varchar(100)	utf8mb4_unicode_ci		Yes	NULL		
9	<b>address</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
10	<b>postcode</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
11	<b>role</b>	tinyint(1)			No	0		
12	<b>created_at</b>	timestamp			Yes	NULL		
13	<b>updated_at</b>	timestamp			Yes	NULL		

### Animals Table Constraints:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b>	bigint(20)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>name</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
3	<b>dob</b>	date			No	None		
4	<b>description</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
5	<b>image</b>	longblob			No	None		
6	<b>availability</b>	enum('Available', 'Unavailable')	utf8mb4_unicode_ci		No	Available		
7	<b>type</b>	enum('Dog', 'Cat', 'Aquarium', 'Bird', 'Mammal', '...')	utf8mb4_unicode_ci		No	Dog		
8	<b>created_at</b>	timestamp			Yes	NULL		
9	<b>updated_at</b>	timestamp			Yes	NULL		

### Adoption\_Requests Table Constraints:

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	<b>id</b>	bigint(100)		UNSIGNED	No	None		AUTO_INCREMENT
2	<b>animalId</b>	int(11)			No	None		
3	<b>petname</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
4	<b>username</b>	varchar(255)	utf8mb4_unicode_ci		No	None		
5	<b>accepted</b>	enum('Approved', 'Pending', 'Rejected', '')	utf8mb4_unicode_ci		No	Pending		
6	<b>created_at</b>	timestamp			Yes	NULL		
7	<b>updated_at</b>	timestamp			Yes	NULL		