Information about animals that are endangered and needs care.

Animal Conservation System

# Introduction

This is an educational website which provides information about animals that are very likely to become extinct in near future. This web app will not only provide details about the endangered species but also categorize them based on their condition.

This website is mainly used for educating young people about nature, it is hoped that they will grow to have a greater empathy towards the environment and be more proactive when it comes to conservation.

# Technologies Used:

**Front End:** HTML5, CSS3, JavaScript, AJAX

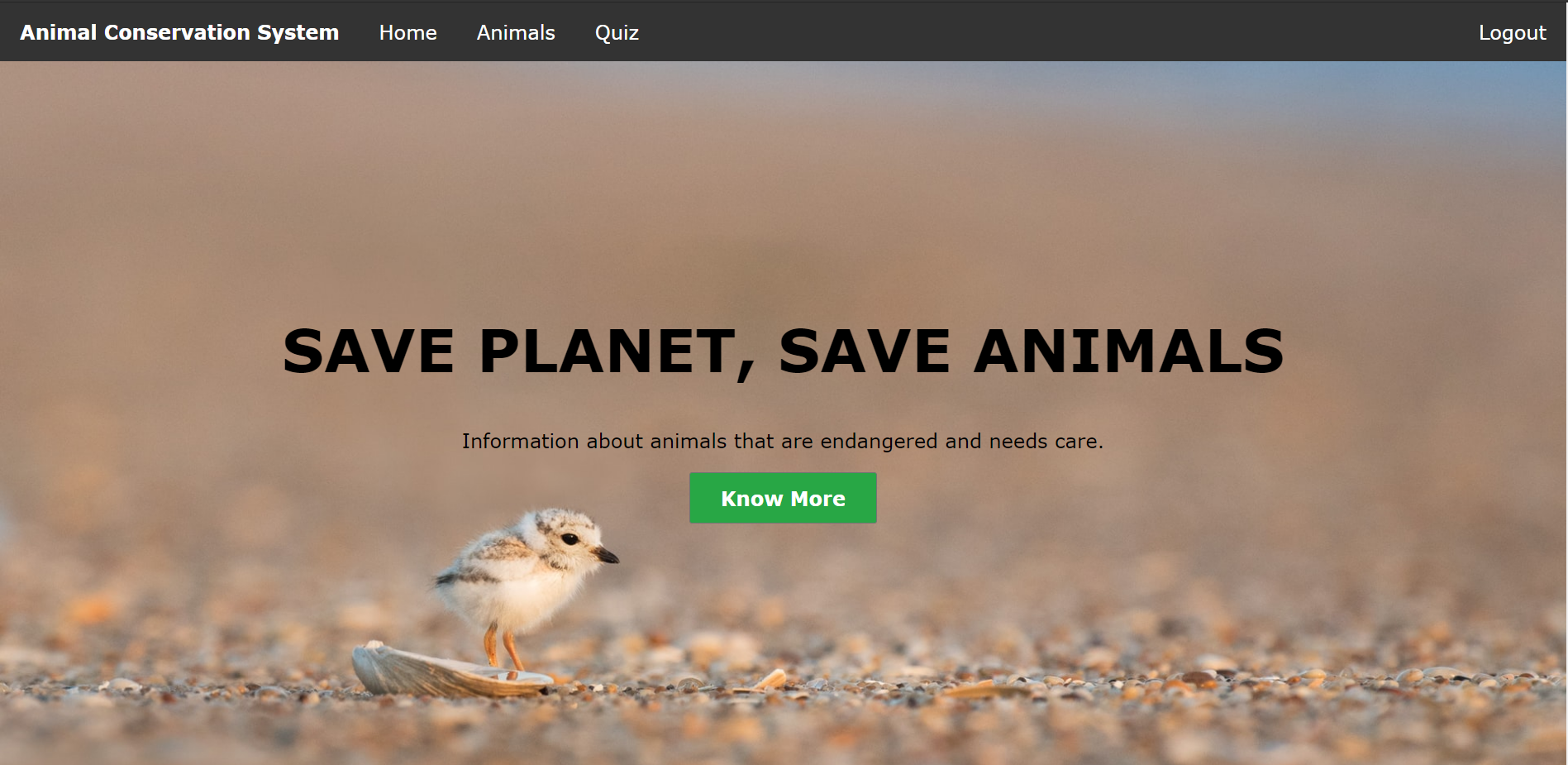
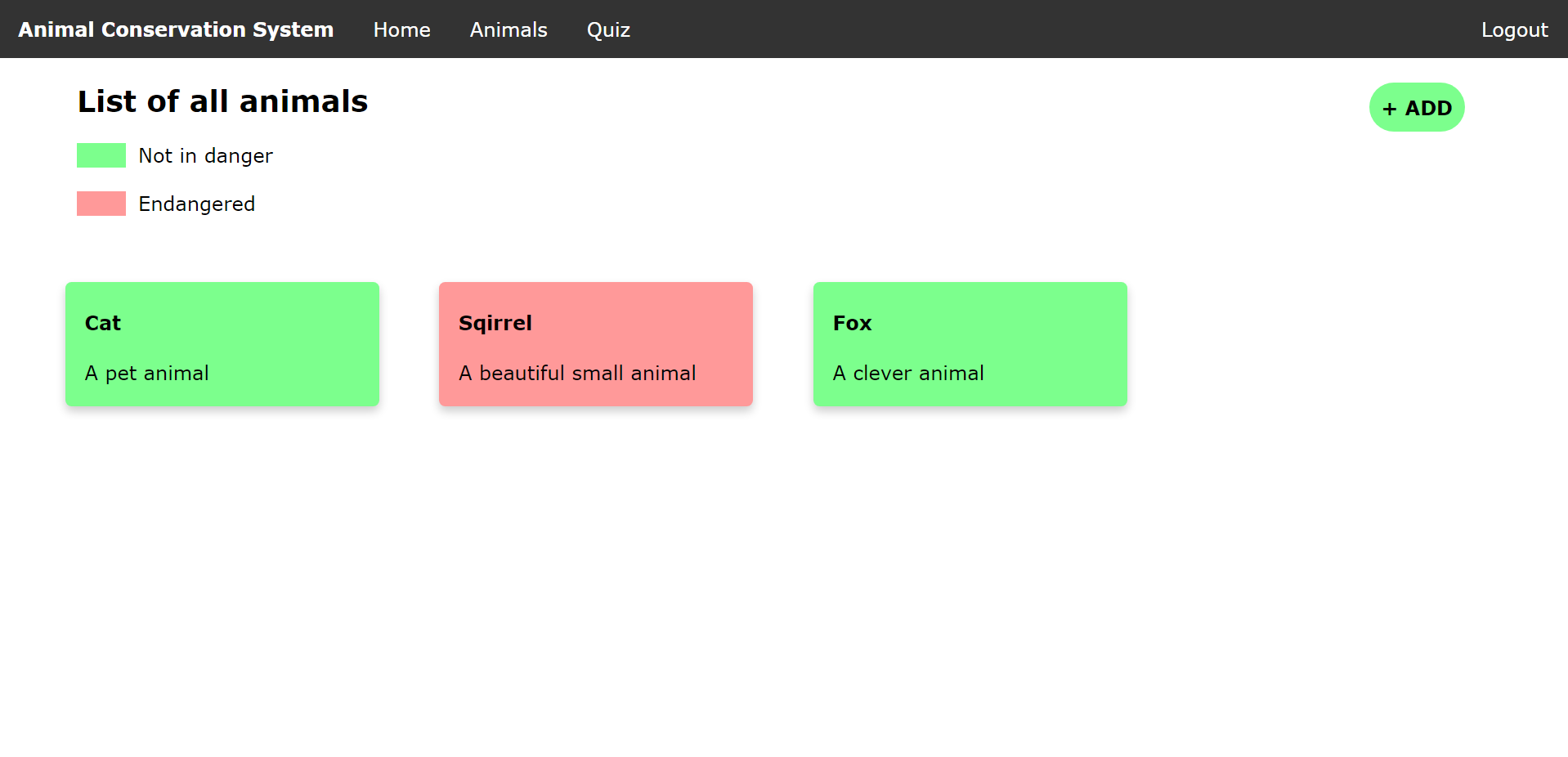
**Backend:** Node JS**,** Express**,** SQLite**,** EJS

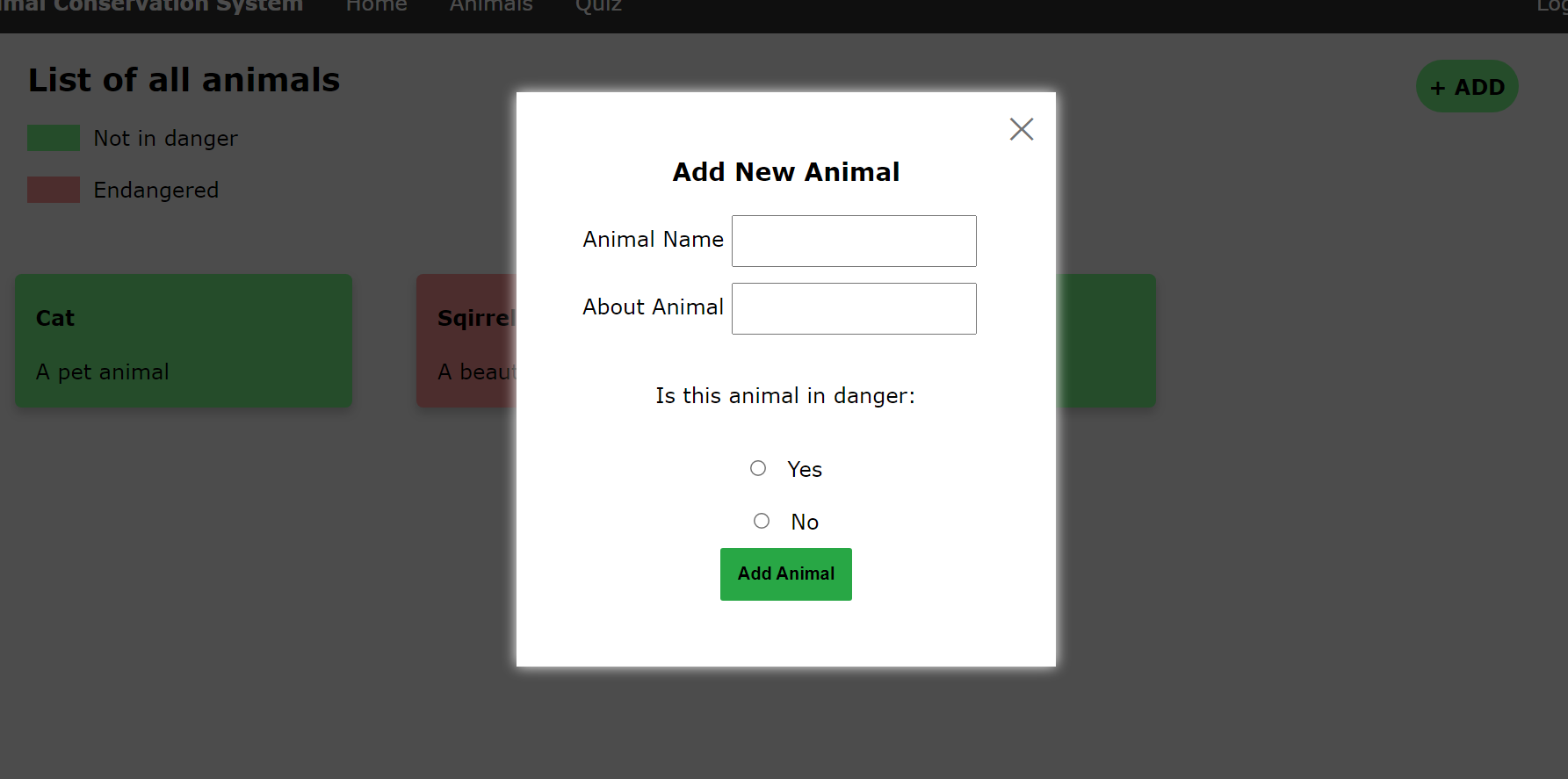
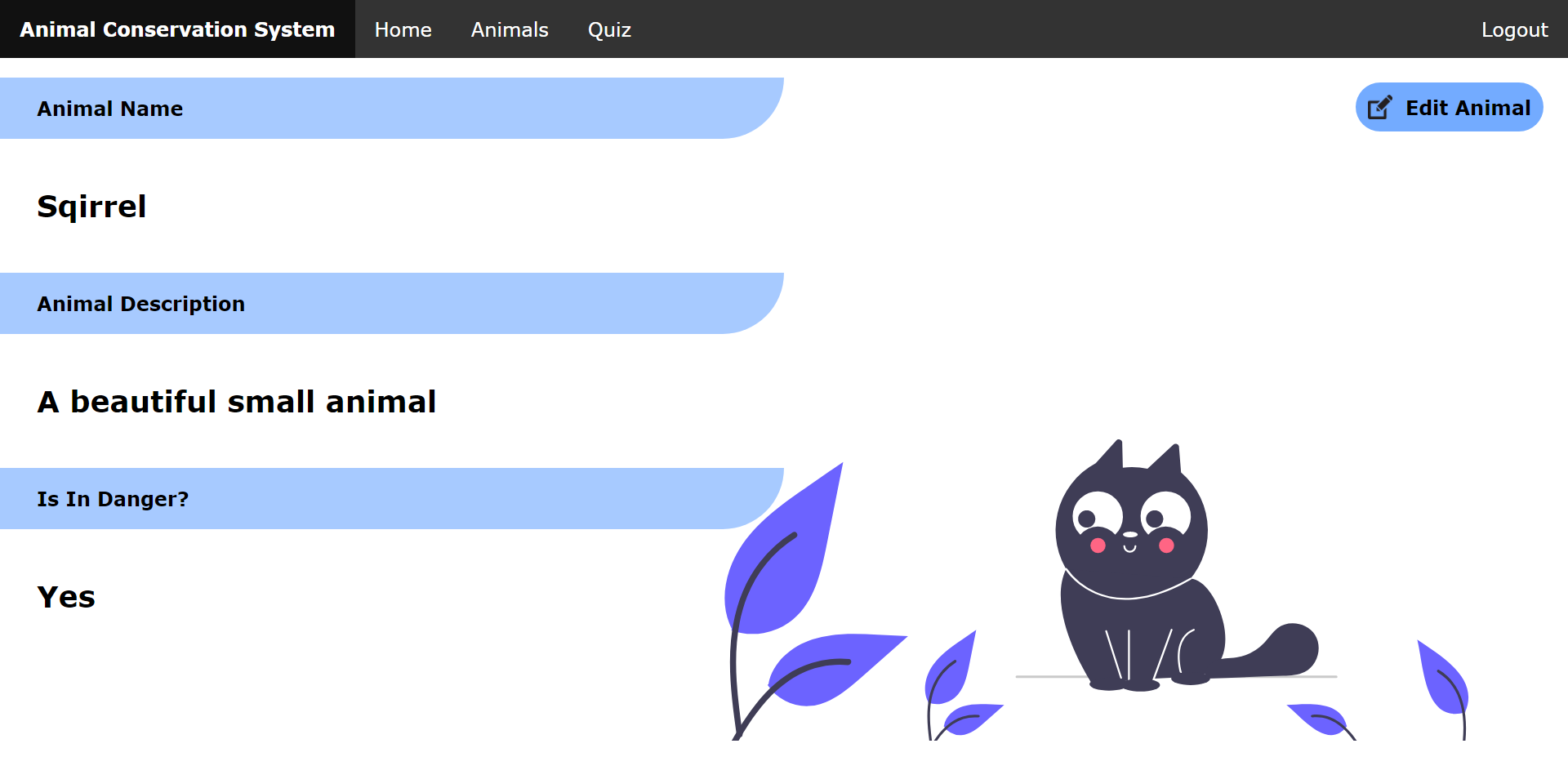
**Database:** sqlite3

**Additional packages:** Passport (For Authentication), Nodemon (For auto reload)

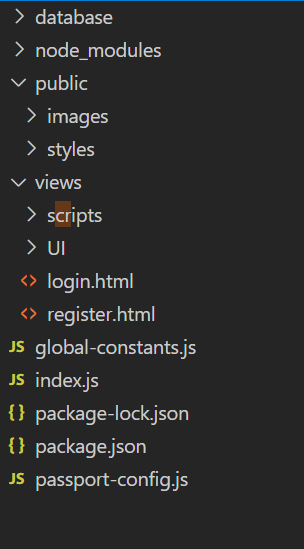
# Highlights:

* Interactive design with **awesome** and responsive UI.

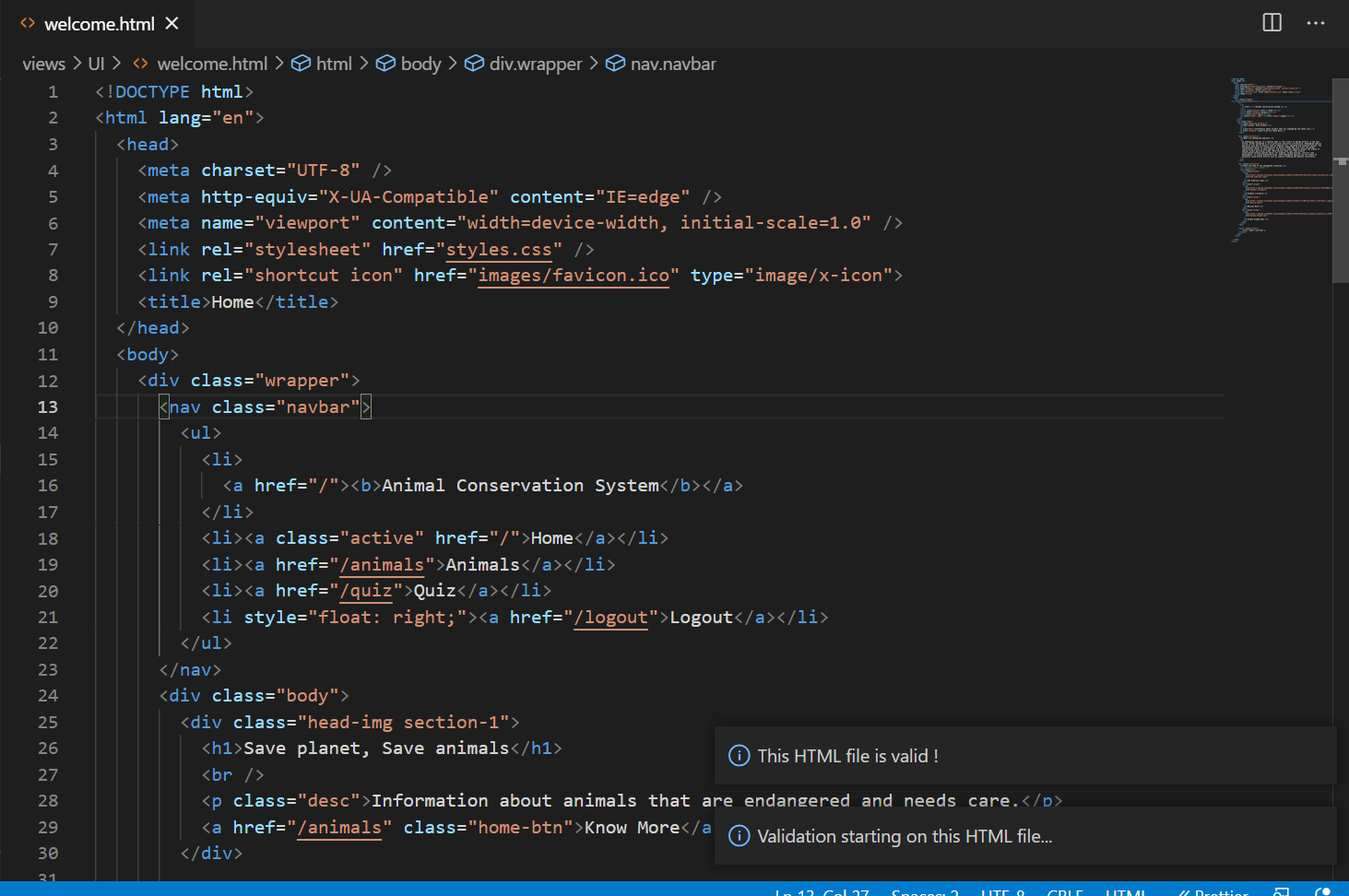
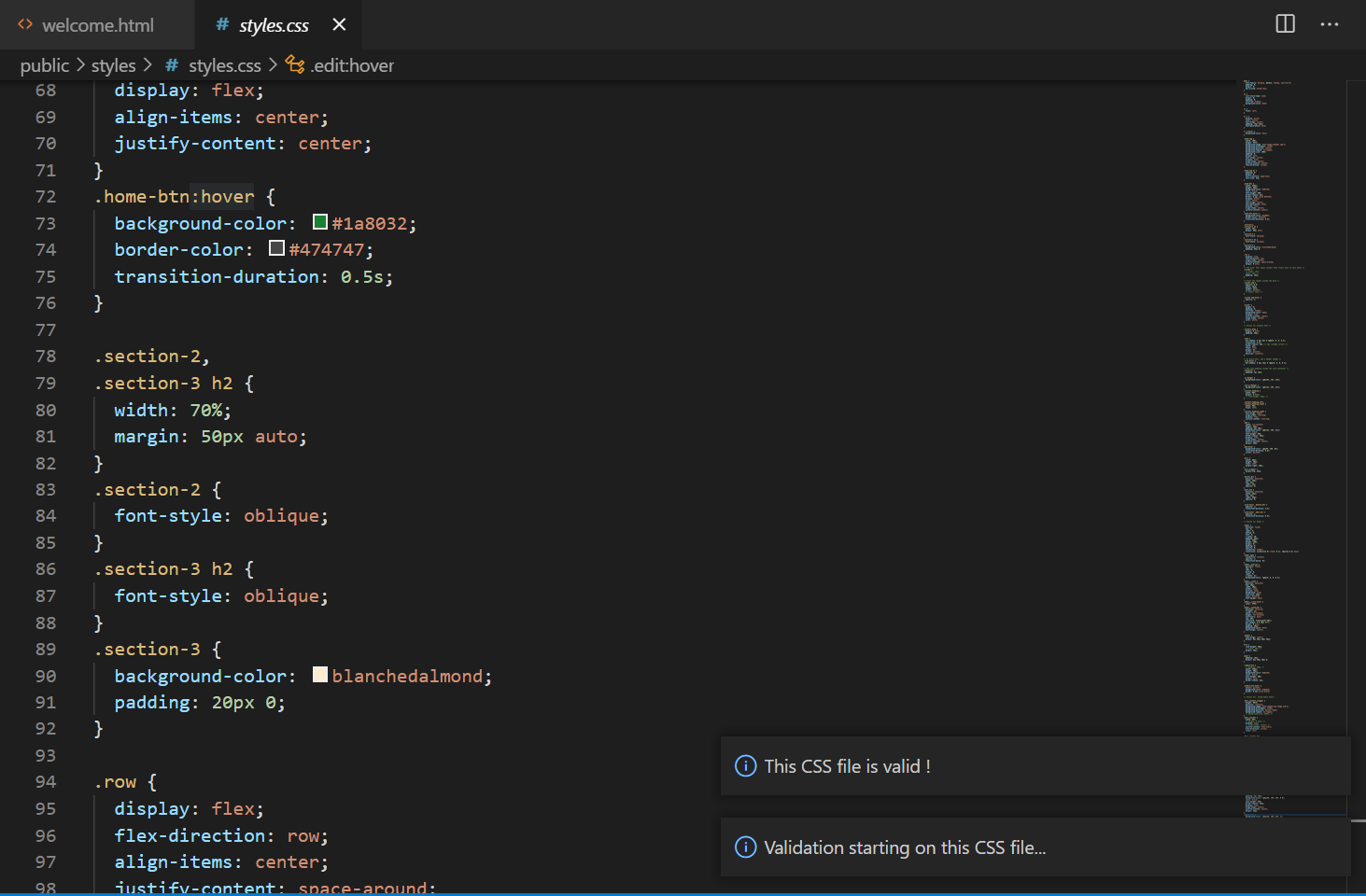
 

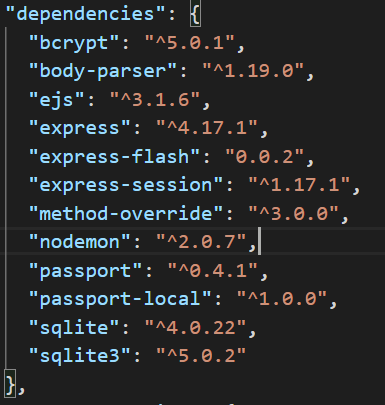
* Followed clear folder structure to understand the project easily.



* Given **comments** wherever required in the code for easy readability.
* Meets all the requirements with **proper** implementation.
* Added **dynamic nature** to HTML elements using Vanilla JavaScript and reduce redundant HTML code for repetitive elements.
* Every HTML and CSS file is validated according to **W3C Standards.**

* HTML forms have **validations** before submitting the data.
* All the packages used are **up to date** with latest versions.



* All the good practices like folder structure, maintaining alt tags in images etc., are followed.
* **Security practices** are followed strictly. For example, used a global constant file for storing SECRET KEYS and other reusable constants across the application. So, if we share the code we can omit sharing that file to the other parties. And encrypting the user entered passwords.

# Features

This website is fulfilled with all the given requirements along with those for “A” Grade. Here is the list of features:

## Backend

* Used NodeJS to create the following API’s with endpoints

|  |  |  |
| --- | --- | --- |
| API | Type | Endpoint |
| User Login | GET | /login |
| User Registration | POST | /register |
| Retrieve a list of all animals | GET | <url>/api/animals |
| Retrieve the information about a specific animal | GET | <url>/api/animals/:id |
| Create a new animal | POST | <url>/api/animals |
| Update the details of an animal | PUT | <url>/ api/animals/:id |
| Remove an animal by its ID | DELETE | <url>/ api/animals/:id |

* Used Passport for authenticating user.
* Used SQLITE3 database (in memory) to store and retrieve data.
* Used Nodemon package for auto reload of the server.
* Have proper validations on backend. For example user cannot add animal with same name again.
* The database schema is as follows (create statements from shell)

CREATE TABLE animals (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, animal\_name NVARCHAR (20) NOT NULL, animal\_desc NVARCHAR (2000) NOT NULL, isEndangered boolean NOT NULL default 0);

CREATE TABLE sqlite\_sequence (name, seq);

CREATE TABLE users (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, username TEXT NOT NULL, email TEXT NOT NULL, password TEXT NOT NULL);

## Frontend

* A production ready application with all the functionalities tested and clean UI.
* The goal of designing this is ease of access to any kind of user with simplicity in mind as well as to include maximum futures possible.
* The following pages are developed corresponding to their API functionalities:

|  |  |  |
| --- | --- | --- |
| Page / URL | Description | Functionality |
| User Login Page (/login) | User login form with email as username and password to authenticate. | Lets user to login to the system. |
| User Registration Page (/register) | User registration form with Name, email and password fields. | Lets user to register himself in database if he visits the website for the first time. |
| Home Page (/) | Contains general information about wildlife, the  purpose of the website and images. | User can land to home page after successful login attempt and can have a glance through the idea of the site. |
| Animals Page (/animals) | Contains the information of all the animals in the database. With proper differentiation between endangered and safe species. User can click on any animal and be redirected to its own description page. Also, user can able to add a new animal and delete it. | Ability to view all the animals with UI representation of endangered and safe species. |
| Able to add new animal, a form will popup with the required fields. You can fill the form and submit to create new animal. |
| Able to redirect to animal specific description page when he clicks on it. |
| User can delete any animal individually by clicking on the delete button on the animal card. |
| He can navigate to any other page from here using navbar. |
| Animal description page (/animal?id=<specificID>) | Page that provides details of an animal with the specified ID. User can click on edit button in this page to edit the animal information. | User can able to visit this page by clicking on specific animal in /animals page. |
| User can able to view the details of that animal in a properly formatted way. |
| Edit Animal option is provided in this page, if the user clicks it a popup with a form with all the prepopulated default values will appear. He can edit the form and submit it to edit the information of that specific animal. |
| Additional Page for higher Grade | | |
| Animal Quiz Page (/quiz) | An interactive animal quiz game created using pure JavaScript. One can start the game, read the questions and select the options. | User can have some fun time answering the questions b choosing the answers from given multiple choice questions. After he choose an answer, he will get a visual feedback if that answer is right or wrong. We can reset the game and also add as many questions as we can in future easily. |

# Legal and ethical considerations

* Accessibility is a part of the design process focused on the user experience, making websites usable by the widest range of people (using the any device) possible.
* The UI of the website is very simple, clean and onto the point.
* There is a proper order to navigate in UI.
* Home -> See All animals -> Add/Delete -> Navigate to animal description page -> Edit an animal.
* User can navigate from one place to another within few clicks.
* Used ALT text for images so that visually impaired users can able to access it.
* Use Sufficient Color Contrast Between Text and Background.
* No legal problem as I used all the images form royalty free websites.
* All images are downloaded from unsplash.com where all photos can be downloaded and used for free for Commercial and non-commercial purposes. No permission needed.
* No script is taken from internet directly.
* All scripts are tested and safe to use.

# Security Considerations

Security is on top priority.

* We have added password encryption for login and register forms. So that only the user will know the password and no one else can read it.
* There are no open api’s from our backend. Everything is authentication protected.
* All the routes/pages of the websites are inaccessible until the user successfully logs in.

# Running the Project

To start the project, make sure you have Latest NodeJS and SQLITE3 are installed in your pc.

Then go to the root directory of the project and run these commands:

npm install – to install packages

npm start – to start the server

open: http://localhost:3000/

# References

For Images: <http://unsplash.com/>

For Home content: <https://en.wikipedia.org/wiki/Endangered_species>

For authentication: <http://www.passportjs.org/> and <https://www.npmjs.com/package/passport>

# Conclusion

The project is delivered in time with all the features and requirements matched.