**Zoo Management in Amazon Web Services**

**Abstract**

The Zoo Management System is a comprehensive web-based platform developed with the primary aim of promoting awareness about endangered wildlife species and facilitating the efficient digital management of animal-related information within a virtual zoo environment. This system serves as both an educational resource and an administrative tool, offering a user-friendly interface that allows visitors to explore detailed profiles of various animal species. These profiles include scientific classifications, habitat information, conservation statuses, behavioral traits, dietary needs, and current efforts being made to protect them.

The platform categorizes animals based on species, regions, and conservation priorities, making it easy for users to navigate and gain insights into specific wildlife groups. In addition to static data, the system incorporates rich visual content such as images and videos, interactive maps, and pop-up windows that provide an immersive learning experience. It also features tools for virtual zoo management, including tracking animal health records, feeding schedules, and enclosure details.

By integrating educational interactivity with backend management capabilities, the Zoo Management System bridges the gap between wildlife conservation efforts and public engagement. It is designed to be scalable and adaptable, making it suitable for use by zoos, wildlife organizations, educational institutions, and the general public. Ultimately, the system aims to foster a deeper appreciation for biodiversity and encourage proactive participation in conservation initiatives through technology-driven awareness.

**Introduction**  
The Zoo Management System is an innovative and educational web-based platform developed to raise awareness about endangered animal species across various biological categories, including mammals, birds, reptiles, aquatic life, and insects. In an age where biodiversity is rapidly declining due to factors such as habitat destruction, climate change, and poaching, this system aims to serve as a digital hub for spreading crucial information about wildlife conservation.

The platform provides an intuitive and engaging user experience through the use of interactive modals, where users can explore detailed profiles of different species. Each modal presents essential information such as the species' common and scientific names, geographic distribution, natural habitat, current population status, and the threats they face. Furthermore, it highlights ongoing conservation measures and offers suggestions on how individuals can contribute to protecting these species.

By combining educational content with interactive design, the Zoo Management System encourages users of all ages to actively engage with the subject of wildlife conservation. Whether accessed by students, educators, researchers, or animal enthusiasts, the system serves as a powerful tool for building awareness and fostering a sense of responsibility toward preserving Earth’s rich and diverse ecosystems.

**Objective of the Problem**  
The primary objective of this project is to design and develop an informative, visually engaging, and user-friendly website that effectively promotes awareness about endangered wildlife species and the urgent need for their conservation. The system aims to bridge the gap between scientific information and public understanding by presenting data in an accessible and interactive format.

Key goals include:

**Education and Awareness**: To inform users about various endangered species across categories such as mammals, birds, reptiles, aquatic life, and insects, highlighting their characteristics, habitats, and current conservation status.

**Visual Engagement**: To utilize images, interactive modals, and multimedia elements that captivate users and enhance their learning experience.

**Ease of Navigation**: To provide a clean and intuitive interface that allows users to quickly access information and explore species by category or conservation status.

**Promotion of Conservation Efforts**: To raise public consciousness about the threats facing wildlife and encourage participation in ongoing conservation initiatives.

**Digital Accessibility**: To ensure the platform is accessible across devices and user groups, making critical information widely available to schools, zoos, environmental organizations, and the general public.

**Hardware and Software Specification**

### ****Hardware Requirements:****

**Processor:** Intel Core i3 or higher (or equivalent AMD processor)

**RAM:** Minimum 4 GB (8 GB recommended for development purposes)

**Storage:** At least 500 MB of free disk space for project files and dependencies

**Display:** Standard HD resolution (1366x768 or higher)

### ****Software Requirements:****

**Operating System:** Compatible with Windows, Linux, or macOS

**Web Browser:** Latest version of any modern browser such as:

Google Chrome

Mozilla Firefox

Microsoft Edge

**Development Tools:**

Visual Studio Code (VS Code)

Sublime Text (alternative)

**Optional Tools and Libraries:**

Git for version control

Node.js (if using JavaScript frameworks)

Live Server extension (for real-time previews)

**Software Features**

The Zoo Management System is designed with a focus on user engagement, educational value, and modern web standards. The software integrates several key features to enhance both functionality and user experience. Below are the core features of the system:

### ****1. Responsive User Interface****

The platform is built with a fully responsive design, ensuring that the user interface adjusts seamlessly across all devices, including desktops, tablets, and smartphones. This makes the system accessible to a wider audience without compromising on visual appeal or usability.

### ****2. Categorization by Species Type****

To simplify navigation and provide structured access to information, animals are categorized into distinct groups such as mammals, birds, reptiles, aquatic life, and insects. This categorization allows users to explore species within specific biological groups and helps in organizing conservation data in an intuitive manner.

### ****3. Interactive Modal Popups for Detailed View****

Each animal profile is displayed through an interactive modal popup, which provides an in-depth view of the species without redirecting the user to a new page. These modals include key details such as the scientific name, conservation status, habitat, physical characteristics, images, and ongoing preservation efforts. This interactive element enhances user engagement and information accessibility.

### ****4. Mobile-Friendly Design****

The system is optimized for mobile devices, ensuring smooth functionality, fast loading times, and easy navigation on smaller screens. Touch-friendly components and simplified layouts are used to improve the experience for mobile users, making the platform convenient to use anytime, anywhere.

### ****5. FontAwesome and Google Fonts Integration****

To enrich the visual aesthetics and iconography of the site, the system uses **FontAwesome** for icons and **Google Fonts** for stylish and readable typography. This integration not only enhances the visual appeal but also maintains consistency and professionalism across the platform.

### ****6. Lightweight and Fast Loading****

Built with clean HTML, CSS, and JavaScript (or additional frameworks, if applicable), the platform is optimized for speed and performance. Compressed assets, asynchronous loading, and minimal use of heavy libraries ensure a smooth user experience with minimal delays.

### ****7. Cross-Browser Compatibility****

The system is tested for compatibility with all major web browsers, including Chrome, Firefox, Edge, and Safari. This ensures that users have a consistent experience regardless of the browser they prefer.

### ****8. Scalability and Extensibility****

The system is designed with scalability in mind, allowing for future enhancements such as user login features, admin control panels, search functionality, multi-language support, or even integration with databases for dynamic content loading.

## ****MODULES****

The Zoo Management System integrates a variety of front-end technologies and cloud computing services to deliver a responsive, media-rich, and highly interactive digital platform. The system has been modularly designed for scalability, performance, and ease of maintenance. Each module is described below with detailed functionality and implementation strategy.

### ****1. HTML (HyperText Markup Language)****

HTML is the structural language of the World Wide Web and serves as the backbone of this system. It is used to define the content hierarchy, elements, and metadata for each page in the application.

Semantic tags like <header>, <nav>, <section>, <footer>, and <article> provide meaning to the structure, aiding both accessibility and SEO.

List elements (<ul>, <ol>, <li>) are used for displaying categories or itemized conservation facts.

<div> containers are used to group content and are styled dynamically using class attributes.

Anchor tags (<a>) facilitate internal linking and navigation between sections, allowing users to explore species categories efficiently.

Multimedia elements like <img>, <video>, and <iframe> are embedded directly within pages to support rich content delivery.

HTML forms are also included for collecting user feedback or interacting with future features such as newsletters or user suggestions.

### ****2. CSS (Cascading Style Sheets)****

CSS is responsible for the design and visual presentation of the HTML elements. It enhances user experience by providing aesthetically pleasing layouts, consistent themes, and adaptable interfaces.

Custom styling is applied using both internal and external stylesheets, ensuring modular and reusable code.

Flexbox and CSS Grid are used extensively to manage complex layouts, such as card-based species displays and responsive navigation bars.

Media queries allow for responsive design, adjusting layouts based on screen size and orientation.

CSS animations and transitions (e.g., on hover or click) add smooth interactions for buttons, modals, and image displays.

The use of pseudo-classes like :hover, :focus, and :nth-child() enables advanced styling with minimal JavaScript.

Google Fonts integration ensures that professional, clean, and readable typography is maintained across all pages.

### ****3. JavaScript****

JavaScript brings dynamic behavior to the website by adding interactivity and client-side logic to the user interface.

Event listeners detect user interactions such as clicks, scrolls, and keyboard inputs, triggering specific behaviors like opening a modal or filtering results.

Functions are used to open and close modal windows that display detailed species data without refreshing the page.

DOM manipulation allows dynamic updates to content sections, such as rendering animal data from a predefined list or API response.

Validation scripts can ensure the accuracy of data entered in contact or feedback forms.

Lightweight JavaScript libraries or Vanilla JS techniques are applied to maintain performance and minimize dependencies.

JavaScript plays a key role in handling asynchronous data loading (AJAX), preparing the project for future integration with backend databases or APIs.

### ****4. AWS (Amazon Web Services)****

AWS provides the cloud infrastructure required for deployment, data management, and scalability of the system. It ensures reliability, global access, and high-performance delivery of the application.

Hosting services like **Amazon EC2** or **AWS Amplify** allow the website to be deployed to the cloud with auto-scaling capabilities.

CloudFront (CDN) can be used for faster content delivery by distributing static files closer to users.

AWS Lambda may be incorporated to handle serverless backend functions such as email notifications, analytics, or API responses.

Amazon Route 53 enables DNS management and custom domain routing for a professional-grade web experience.

AWS CloudWatch and CloudTrail can monitor usage, detect anomalies, and maintain operational logs for better security and performance auditing.

AWS architecture enables zero-downtime deployments, which is critical for educational systems accessed by users at different times globally.

### ****5. AWS S3 (Simple Storage Service)****

Amazon S3 is used for secure, scalable, and high-availability storage of static files and media assets used throughout the Zoo Management System.

Image assets for each animal species are uploaded to specific S3 buckets and accessed using pre-signed or public URLs.

Objects are stored in a flat, key-based hierarchy which simplifies file retrieval and integration with front-end content.

Bucket policies and Access Control Lists (ACLs) define the level of visibility—private, public-read, or restricted.

Versioning features are enabled to maintain backup copies and track changes over time for files like icons or PDFs.

Lifecycle configuration automates data archiving, deletion, or transition to lower-cost storage tiers for long-term efficiency.

Static website hosting on S3 can serve entire HTML/CSS/JS pages directly from buckets when combined with Route 53 and CloudFront.

### ****6. Access Management****

Access Management defines how resources are protected from unauthorized use. It encompasses mechanisms to authenticate users and authorize their interactions within the system.

Logical roles such as Admin, Viewer, and Contributor are pre-defined to ensure separation of privileges.

Access control mechanisms are enforced using front-end logic and back-end IAM policies.

Controls are applied at both file-level (e.g., image upload rights) and action-level (e.g., who can modify species information).

Secure headers and form tokens are used to prevent common threats like Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF).

In multi-user environments, access management allows the delegation of responsibilities without compromising overall system security.

Admin dashboards or content management tools (future scope) can be restricted based on access roles.

### ****7. AWS IAM (Identity and Access Management)****

AWS IAM is responsible for managing identities, credentials, and permissions within the AWS ecosystem used by the Zoo Management System.

IAM Users are created for each developer, admin, or automation script to ensure traceable and secure interactions.

IAM Roles are used to grant temporary permissions to AWS services such as Lambda or EC2 instances without exposing credentials.

IAM Policies are written in JSON format and define fine-grained access to services like S3, CloudWatch, and Amplify.

Multi-Factor Authentication (MFA) enhances login security for sensitive AWS accounts.

Permission boundaries are enforced to prevent overly permissive access across the development and production environments.

Logs of user actions and changes to IAM policies are automatically tracked through AWS CloudTrail for compliance monitoring.

## ****SYSTEM ANALYSIS****

The Zoo Management System is a front-end-focused web application designed to promote awareness about endangered wildlife species through an engaging, educational, and visually rich interface. Since the system does not incorporate a backend database or server-side logic, the core analysis revolves around its **user interface (UI)**, **user experience (UX)**, **design modularity**, and **technology integration**.

### ****1. User Interface (UI) Design Analysis****

The UI is crafted using standard web technologies including HTML, CSS, and JavaScript. The design prioritizes clarity, accessibility, and ease of navigation:

**Navigation Bar:** Offers quick access to various categories such as mammals, birds, reptiles, insects, and aquatic species.

**Species Cards:** Each species is visually represented using interactive cards that include thumbnails, names, and short descriptions.

**Modal Popups:** When a card is clicked, a detailed modal appears, presenting extended information about the species, its habitat, and conservation status.

**Consistent Theme:** Use of a consistent color palette, iconography (via FontAwesome), and typography (via Google Fonts) ensures visual coherence across pages.

### ****2. User Experience (UX) Design Analysis****

The application prioritizes an intuitive and user-friendly experience:

**Responsive Design:** The interface adapts seamlessly to different screen sizes including desktops, tablets, and mobile devices.

**Performance Optimization:** Lightweight front-end code and media handling via cloud storage (AWS S3) ensure fast loading times and smooth transitions.

**Educational Interactivity:** Modal windows act as mini-information hubs, enabling users to explore facts without leaving the current page.

**Touch-Friendly Design:** Interactive components are optimized for touch gestures, enhancing mobile usability.

### ****3. Modularity and Scalability****

Even in the absence of a backend, the system is designed in a modular and extensible manner:

**Component-Based Structure:** Reusable components such as cards, buttons, and modals make it easy to expand or reorganize content.

**Separation of Concerns:** HTML handles structure, CSS manages styling, and JavaScript adds logic—each layer is clearly defined and independently maintainable.

**Future Integration Ready:** The codebase is organized in such a way that it can easily accommodate backend services (e.g., APIs or databases) in the future if desired.

### ****4. Cloud Integration Analysis****

Although backend logic is not implemented, cloud services are integrated for asset management and deployment:

**AWS S3:** Used to host and deliver images and other static media assets. The use of cloud storage ensures high availability, durability, and fast access.

**AWS IAM:** Access to cloud resources is securely managed using IAM policies and roles, ensuring proper permission control during development and deployment.

### ****5. Security Considerations****

While server-side authentication and database security are not required in this version, basic front-end best practices are applied:

**Sanitized Markup:** All HTML and JavaScript code is written to prevent injection vulnerabilities.

**Access Control via IAM:** AWS assets are protected using IAM policies and bucket configurations to restrict unauthorized access.

### ****6. Limitations****

**No Database or Server-Side Logic:** Limits functionality such as user logins, data storage, or real-time interaction.

**Static Data:** Information must be hardcoded or manually updated unless extended with dynamic data fetching from APIs or cloud functions.

## ****EXISTING SYSTEM****

Traditional zoo websites and wildlife awareness platforms often serve as static informational portals, offering limited interactivity and outdated design standards. While these platforms provide basic details about animal species, they frequently fall short in terms of user engagement, accessibility, and educational effectiveness—especially for younger audiences or mobile users.

### ****Key Limitations of Existing Systems:****

**Lack of Interactivity**

Most conventional zoo websites present text-heavy content with minimal user engagement.

Species information is often displayed in long paragraphs without visual or interactive aids.

Users are not encouraged to explore or learn through dynamic content such as modals, videos, or quizzes.

**Outdated UI/UX Design**

Many existing platforms use dated layouts that are not aligned with modern design trends.

Visual clutter, poor color contrast, and inconsistent styling make navigation difficult and less appealing.

Fonts, icons, and layout structures often fail to deliver an enjoyable browsing experience.

**Poor Mobile Responsiveness**

A large portion of existing wildlife-related websites are not optimized for mobile devices.

Layouts break, content overflows, and interactive elements become unusable on smaller screens.

With increasing mobile traffic, this lack of responsiveness leads to poor accessibility and user retention.

**Limited Categorization and Filtering**

Existing websites usually present species data in bulk without clear categorization or filtering options.

Users may find it difficult to explore animals by type (e.g., mammals, reptiles) or conservation status.

Lack of search functionality or filtering reduces educational efficiency and user satisfaction.

**Minimal Use of Multimedia**

Multimedia content like high-resolution images, videos, audio clips, or interactive maps is often missing or underutilized.

This results in a less immersive experience, particularly when educating users about endangered species or natural habitats.

**Static and Hardcoded Content**

Most content is manually updated and lacks any form of real-time integration with external conservation databases or news.

The absence of backend integration or dynamic APIs limits the freshness and relevance of information.

## ****PROPOSED SYSTEM****

The proposed system is a modern, interactive, and educational web-based platform that addresses the limitations of traditional zoo and wildlife websites. It is specifically designed to spread awareness about endangered animal species across various categories, while ensuring a seamless user experience across all devices.

### ****Key Features and Improvements:****

1. **Interactive and Engaging Interface**
   * The platform utilizes responsive design principles, allowing users to explore animal species through interactive components such as cards and modals.
   * Each animal profile includes images, species name, habitat, conservation status, and additional facts presented in an appealing popup format.
   * Animations and transitions enhance engagement, making the learning process more enjoyable and memorable.
2. **Categorized Display of Species**
   * Animals are organized into intuitive categories such as mammals, birds, reptiles, insects, and aquatic life.
   * This structure allows users to easily browse based on interest or educational purpose.
   * Clear labeling and filters improve accessibility and help users quickly find relevant species information.
3. **Mobile-Responsive Design**
   * The platform is fully optimized for smartphones, tablets, and desktops.
   * Adaptive layouts ensure content is accessible and well-formatted regardless of screen size.
   * Touch-friendly interactions make navigation and modal exploration smooth and user-friendly on mobile devices.
4. **Educational Focus**
   * Each species section provides not only biological details but also highlights the reasons for endangerment and ongoing conservation efforts.
   * The system aims to raise awareness and educate users about the importance of biodiversity and ecosystem balance.
   * Simple language and visual aids make it suitable for students, researchers, and the general public alike.
5. **Modern Web Technologies**
   * Built using HTML, CSS, and JavaScript, the system leverages cloud services such as AWS S3 for efficient asset delivery.
   * Modularity in design allows for future enhancements, such as backend integration, live updates from conservation APIs, or user-submitted content.
6. **Lightweight and Fast Performance**
   * The site is optimized to load quickly, even on slower connections, due to minimal dependencies and cloud-hosted static content.
   * Optimized images and code ensure that users can interact with content without unnecessary delays.

## ****DATA FLOW DIAGRAM (DFD)****

Data Flow Diagrams (DFDs) are used to visually represent the flow of information within a system. Since the Zoo Management System is a **static website** without backend logic, the data flow is limited to **client-side interactions**. However, DFDs still help in understanding how users interact with the system components.

### ****DFD Level 0 – Context-Level Diagram****

This level represents the overall system as a single process, showing the interaction between the user and the Zoo Website.

#### **Components:**

* **External Entity:** User
* **Process:** Zoo Website
* **Data Flow:** View Species
* **Data Store:** Not applicable (no persistent storage)

#### **Description:**

At Level 0, the user interacts with the Zoo Website by accessing it through a browser and viewing the static content about animal species.

#### **Diagram (Conceptually):**

sql

CopyEdit

+-------+ +------------------+

| User | -----> | Zoo Website |

+-------+ +------------------+

<------------ View Species

### ****DFD Level 1 – Functional Decomposition****

Level 1 breaks down the single process from Level 0 into more detailed sub-processes, showing how the system components handle user interaction.

#### **Components:**

1. **Process 1.1:** View Category
2. **Process 1.2:** Select Species
3. **Process 1.3:** View Modal Information
4. **External Entity:** User
5. **Data Stores:** Static HTML/JS Content

#### **Data Flow:**

* User selects a category (e.g., mammals, birds)
* The system displays species in that category
* The user clicks a specific species
* A modal popup appears with detailed information

#### **Diagram (Conceptually):**

sql

CopyEdit

+-------+

| User |

+-------+

|

v

+----------------+

| View Category | <------+

+----------------+ |

| (Static HTML/JS)

v

+----------------+

| Select Species |

+----------------+

|

v

+------------------------+

| View Modal Information |

+------------------------+

|

v

(Conservation Details, Image, Facts)

## ****USE CASE DIAGRAM****

The use case diagram describes how users interact with the Zoo Management System. Since the system is a static website with client-side interactions only, the functionality revolves around user navigation and content viewing.

### ****Primary Use Case: View Species Information****

#### **Actor:**

* **Website User** – The end-user who visits the website to learn about endangered species.

#### **System:**

* **Zoo Website** – A static web application built using HTML, CSS, JavaScript, and cloud-hosted assets.

#### **Use Case Description:**

The user browses the website and selects a species category (e.g., mammals, birds). Upon clicking a specific species, a modal popup opens, displaying detailed information such as the species name, image, habitat, and conservation measures.

#### **Use Case Flow:**

1. **User opens the website** on a desktop or mobile browser.
2. **User selects a species category** from the homepage navigation.
3. **User clicks on a species card** within the chosen category.
4. The system **displays a modal popup** with detailed species information.
5. **User reads the content** and closes the modal.

### ****Textual Representation:****

| **Actor** | **Use Case** | **System Response** |
| --- | --- | --- |
| Website User | View Species Info | Display species modal with details |

### ****Diagram Description:****

Here’s a conceptual visualization of the use case:

pgsql

CopyEdit

+-------------------+

| Website User |

+-------------------+

|

| Initiates

v

+-------------------+

| View Species Info|

+-------------------+

|

v

+--------------------------+

| Display Species Modal |

+--------------------------+

### ****Extended Use Cases (Optional for Future Enhancements):****

If the system is expanded with backend or interactive features, additional use cases might include:

* Submit Feedback
* Save Favorite Species
* Register/Login
* Admin: Add/Remove Species

## ****SYSTEM DESIGN****

The Zoo Management System is designed as a static, front-end-only web application that focuses on delivering an informative and visually rich user experience. The architecture is structured into multiple layers using HTML, CSS, and JavaScript, supported by web assets like fonts and icons to enhance aesthetics and interactivity.

### ****1. HTML – Structural Design****

* HTML (HyperText Markup Language) forms the **backbone of the website**, defining the structure and layout of all web elements.
* Semantic tags like <section>, <nav>, <main>, and <article> are used to organize content logically and improve accessibility.
* Each species is represented using consistent components like cards or tiles, grouped under categories (e.g., mammals, birds).
* Modal structures are defined using <div> elements, hidden by default and triggered via JavaScript.

#### Example Elements:

* Header with logo and navigation menu
* Category sections with species thumbnails
* Modal containers for detailed views

### ****2. CSS – Styling and Responsiveness****

* CSS (Cascading Style Sheets) provides **styling, color schemes, and responsive layout design**.
* A **mobile-first approach** is followed, ensuring optimal experience across all devices—phones, tablets, and desktops.
* Flexbox and Grid Layouts are used to manage content alignment and spacing.
* Media queries adjust content display based on screen width, improving usability on smaller screens.
* Custom classes ensure consistent margins, paddings, typography, and hover effects across components.

#### Features:

* Responsive navigation bar
* Card hover animations
* Modal transitions and overlays
* Theming with custom colors and gradients

### ****3. JavaScript – Interactivity and Logic****

* JavaScript enables **dynamic content behavior** such as modal handling and category switching.
* On clicking a species card, a script triggers the modal to appear and loads the appropriate content.
* Event listeners are used for:
  + Opening and closing modals
  + Navigating between sections
  + Displaying tooltips or additional info
* Lightweight scripts are written to maintain high performance with minimal dependencies.

#### Example Features:

* Modal popup management
* Scroll-to-category functionality
* Toggle buttons for theme/lightbox effects

### ****4. Aesthetic Enhancements****

#### **FontAwesome Icons**

* Used for visual cues and icons (e.g., animal categories, navigation arrows, close buttons).
* Enhances usability and adds a modern look to the UI.

#### **Google Fonts**

* Applied for clean, readable, and attractive typography.
* Helps maintain a consistent visual style throughout the site.

### ****5. Optional Enhancements (Future Scope)****

* Integration with a backend for dynamic content
* Addition of animations with libraries like AOS or GSAP
* Cloud-based asset delivery via AWS S3 for faster loading

## ****INPUT DESIGN****

In traditional web systems, input design refers to the planning and implementation of how data is entered, validated, and processed within an application. However, the **Zoo Management System** is a static, front-end-only website with **no form-based or backend data entry mechanisms**.

Despite the absence of textual or form-based inputs, the system does accept **user interactions through clicks and selections**, which are processed on the client side using JavaScript.

### ****Nature of Input in This System****

Since there is no user registration, login, or form submission, the **only input mechanism** involves **mouse or touch interactions** with the UI elements.

### ****Types of Inputs Used:****

| **Input Method** | **Description** | **System Response** |
| --- | --- | --- |
| Click / Tap | Clicking on a species card or icon | Triggers modal popup showing species information |
| Hover | Hovering over cards or buttons (on desktop) | Reveals animations or tooltips |
| Scroll | Scrolling through the page | Loads more content in view (visually) |
| Navigation Click | Clicking on menu links or category buttons | Scrolls or jumps to the relevant section |
| Modal Close | Clicking on the close icon or background overlay | Hides the modal popup |

### ****Input Handling via JavaScript:****

* JavaScript listens for **click events** on specific HTML elements using addEventListener.
* Based on the clicked element, it identifies the corresponding content to load into the modal.
* The modal is displayed using style transitions (e.g., display: block, opacity: 1).
* Close buttons or clicking outside the modal reverses this action.

### ****No Form Inputs Present:****

* No <input>, <textarea>, or <form> elements exist in this static implementation.
* As a result, there is:
  + No data validation
  + No server requests
  + No session management or user input storage

## ****SYSTEM TESTING****

System testing is a critical phase in the development lifecycle, used to validate that the website works as intended across different platforms, devices, and environments. For the **Zoo Management System**, testing primarily focused on **responsiveness**, **interactivity**, and **visual consistency**, given that it's a static, front-end-only website.

### ****Testing Objectives:****

* Ensure full compatibility across major browsers.
* Confirm proper rendering and layout on different screen sizes.
* Validate that modal popups and navigation links work without error.
* Ensure that all media and assets (images, fonts, icons) load correctly.

### ****1. Cross-Browser Testing:****

The website was tested on the latest versions of the following browsers:

| **Browser** | **Result** |
| --- | --- |
| Google Chrome | Fully compatible |
| Mozilla Firefox | Fully compatible |
| Microsoft Edge | Fully compatible |
| Safari | Compatible (macOS/iOS) |
| Opera | Fully compatible |
| Brave | Fully compatible |

**Notes:**

* FontAwesome icons and Google Fonts rendered correctly across all browsers.
* CSS transitions and JavaScript modals behaved consistently.

### ****2. Device Compatibility Testing:****

Responsiveness was tested using real devices and browser developer tools.

| **Device Type** | **Operating System** | **Screen Sizes Tested** | **Result** |
| --- | --- | --- | --- |
| Desktop/Laptop | Windows, macOS | 1366x768, 1920x1080 | Fully functional |
| Tablet | Android, iPadOS | 768x1024, 800x1280 | Responsive layout |
| Smartphone | Android, iOS | 360x640, 414x896 | Mobile-friendly |

**Highlights:**

* Navigation bar adapts on smaller screens.
* Modals resize dynamically to fit various viewports.
* Font size and image scaling tested for readability on mobile.

### ****3. Functional Testing:****

Tested all UI interactions:

| **Feature** | **Test Result** |
| --- | --- |
| Category navigation | Passed |
| Modal popup functionality | Passed |
| Modal close button | Passed |
| Scroll behavior | Passed |
| Icon and font loading | Passed |

### ****4. Performance and Load Testing:****

Given that the site is static:

* Pages loaded quickly even with multiple images.
* No lag observed in modals or UI animations.
* Optimized images and use of CDN (e.g., FontAwesome, Google Fonts) helped reduce loading time.

## ****TYPES OF TESTING****

### ****1. UI Testing (User Interface Testing)****

**Purpose:**  
To verify that all user interface elements are correctly aligned, visually appealing, and consistent with the design.

**Key Checks:**

* Proper layout of species cards and categories
* Font consistency using Google Fonts
* Icon rendering via FontAwesome
* Modal popup design and readability
* Color themes and hover effects

**Result:**  
All UI elements appeared clean and consistent on all tested devices.

### ****2. Responsiveness Testing****

**Purpose:**  
To ensure that the website layout adjusts and functions correctly across various screen sizes and device orientations.

**Key Checks:**

* Navigation bar adapts on small screens
* Content scaling on mobile, tablet, and desktop
* Modal resizing and content wrapping
* Image and text alignment in portrait and landscape modes

**Result:**  
Responsive behavior was observed across all screen resolutions and device types.

### ****3. Cross-Browser Testing****

**Purpose:**  
To validate the site's performance and appearance across different web browsers.

**Browsers Tested:**

* Google Chrome
* Mozilla Firefox
* Microsoft Edge
* Safari (macOS & iOS)
* Opera
* Brave

**Key Checks:**

* Font and icon compatibility
* CSS and layout rendering
* Modal animation behavior
* Image loading and scaling

**Result:**  
Website performed consistently across all major browsers.

### ****4. Functional Modal Testing****

**Purpose:**  
To ensure that modal popups behave as expected when users interact with species cards.

**Key Checks:**

* Modal opens with correct content on click
* Modal overlays the page and disables background scroll
* Close button functions properly
* Clicking outside the modal also closes it
* All modals are responsive and readable

**Result:**  
All modal-related interactions worked as intended without any bugs.

## ****TEST CASE REPORT****

| **Test Case ID** | **Action Performed** | **Expected Output** | **Actual Result** | **Status** |
| --- | --- | --- | --- | --- |
| TC1 | Click on a species card | Modal opens with relevant species info | Modal opened correctly with content | Pass |
| TC2 | Click on modal close button | Modal disappears and original page returns | Modal closed as expected | Pass |
| TC3 | Load site on mobile device | Layout adjusts to screen size (responsive) | Layout adapted properly | Pass |
| TC4 | Click on category in navbar | Page scrolls to selected category section | Smooth scroll and correct section shown | Pass |
| TC5 | Load icons and fonts | All icons and fonts display properly | FontAwesome and Google Fonts loaded correctly | Pass |
| TC6 | Hover over species card (desktop) | Visual effect (e.g., shadow or zoom) shown | Hover effect triggered as expected | Pass |
| TC7 | Resize browser window | Content rearranges to fit screen width | Content realigned correctly | Pass |
| TC8 | Click outside modal | Modal closes automatically | Modal dismissed as expected | Pass |
| TC9 | Test on Firefox browser | Full compatibility | Functioned as expected | Pass |
| TC10 | Test on Safari (iOS) | Full compatibility | Functioned as expected | Pass |

### ****Summary:****

* **Total Test Cases:** 10
* **Passed:** 10
* **Failed:** 0
* **System Status:** **Stable and Fully Functional**

SAMPLE CODING

HTML CODE   
  
index.html  
  
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Wildlife Protector</title>

<link href="https://fonts.googleapis.com/css2?family=Poppins:wght@300;400;500;600;700&display=swap" rel="stylesheet">

<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">

<link rel="stylesheet" href="css/styles.css">

</head>

<body>

<!-- All your existing HTML content remains exactly the same -->

<div class="container">

<h1>Protect Our Endangered Wildlife</h1>

<!-- Mammals Section -->

<div class="category-section">

<div class="category-header">

<div class="category-icon">

<i class="fas fa-paw"></i>

</div>

<h2 class="category-title">Mammals in Danger</h2>

</div>

<div class="species-grid" id="mammals"></div>

</div>

<!-- Birds Section -->

<div class="category-section">

<div class="category-header">

<div class="category-icon">

<i class="fas fa-dove"></i>

</div>

<h2 class="category-title">Birds at Risk</h2>

</div>

<div class="species-grid" id="birds"></div>

</div>

<!-- Reptiles Section -->

<div class="category-section">

<div class="category-header">

<div class="category-icon">

<i class="fas fa-sticky-note"></i>

</div>

<h2 class="category-title">Reptiles in Peril</h2>

</div>

<div class="species-grid" id="reptiles"></div>

</div>

<!-- Aquatic Section -->

<div class="category-section">

<div class="category-header">

<div class="category-icon">

<i class="fas fa-fish"></i>

</div>

<h2 class="category-title">Aquatic Life Threatened</h2>

</div>

<div class="species-grid" id="aquatics"></div>

</div>

<!-- Insects Section -->

<div class="category-section">

<div class="category-header">

<div class="category-icon">

<i class="fas fa-bug"></i>

</div>

<h2 class="category-title">Insects in Decline</h2>

</div>

<div class="species-grid" id="insects"></div>

</div>

<!-- Modal -->

<div class="modal" id="modal">

<div class="modal-content">

<button class="modal-close" id="close-modal">

<i class="fas fa-times"></i>

</button>

<img class="modal-image" id="species-img" src="" alt="">

<div class="modal-body">

<h2 class="modal-title" id="species-title"></h2>

<ul class="conservation-list" id="species-info"></ul>

</div>

</div>

</div>

</div>

<script src="js\script.js"></script>

</body>

</html>  
  
  
Js Code   
  
Script.js

const speciesList = [

{

name: "Bengal Tiger",

category: "mammals",

img: "images/tiger.jpeg",

info: [

"Strengthen anti-poaching measures through community-led patrols",

"Establish wildlife corridors between fragmented habitats",

"Promote sustainable tourism practices in protected areas"

]

},

{

name: "African Elephant",

category: "mammals",

img: "images/elephant.jpeg",

info: [

"Implement advanced tracking systems for herd protection",

"Develop alternative livelihoods for communities near habitats",

"Enforce global ban on ivory trade"

]

},

{

name: "Panda",

category: "mammals",

img: "images/Panda.jpeg",

info: [

"Protect bamboo forests, their primary food source, from human expansion and logging.",

"Prevent habitat fragmentation by creating protected green corridors between reserves.",

"Support breeding programs that enhance genetic diversity and help reintroduce pandas into the wild."

]

},

{

name: "Polar Bear",

category: "mammals",

img: "images/Polar Bear.jpeg",

info: [

"Combat climate change by reducing greenhouse gas emissions that melt sea ice.",

"Preserve arctic habitats through marine protected areas and research-driven conservation policies.",

"Avoid oil drilling in the Arctic which disrupts ecosystems and leads to oil spills harmful to wildlife."

]

},

{

name: "Snow Leopard",

category: "mammals",

img: "images/Snow Leopard.jpeg",

info: [

"Monitor highland ecosystems to assess snow leopard populations and their prey base.",

"Support anti-poaching units and install camera traps for real-time tracking of threats.",

"Educate local communities about coexistence and involve them in conservation efforts."

]

},

{

name: "Red Panda",

category: "mammals",

img: "images/Red Panda.jpeg",

info: [

"Preserve Himalayan forest habitats through sustainable land use.",

"Prevent poaching by enforcing wildlife protection laws.",

"Support community-led eco-tourism to generate local conservation income."

]

},

{

name: "Orangutan",

category: "mammals",

img: "images/Orangutan.jpeg",

info: [

"Stop palm oil-driven deforestation in Southeast Asia.",

"Rehabilitate and reintroduce rescued orangutans into the wild.",

"Educate consumers on sustainable palm oil sourcing."

]

},

{

name: "Black Rhino",

category: "mammals",

img: "images/Black Rhino.jpeg",

info: [

"Strengthen anti-poaching surveillance and enforcement.",

"Translocate rhinos to safe, secure reserves with breeding potential.",

"Educate local communities on rhino value to ecosystems and economy."

]

},

{

name: "Parrot",

category: "birds",

img: "images/Parrot.jpeg",

info: [

"Prevent illegal bird trade by raising awareness and enforcing strict customs checks.",

"Preserve rainforests which are essential for nesting and feeding behaviors.",

"Support rescue centers that rehabilitate captured parrots and reintegrate them into the wild."

]

},

{

name: "Owl",

category: "birds",

img: "images/Owl.jpeg",

info: [

"Protect nesting sites in old trees and cliffs to ensure breeding success.",

"Reduce pesticide use that contaminates prey and weakens owl populations.",

"Avoid nighttime disturbances near habitats to maintain healthy nocturnal activity patterns."

]

},

{

name: "Bald Eagle",

category: "birds",

img: "images/Bald Eagle.jpeg",

info: [

"Protect waterways that supply fish, their primary diet.",

"Ban harmful chemicals like DDT that impact reproduction and eggshell strength.",

"Enforce hunting laws to prevent illegal shooting and disturbance of nests."

]

},

{

name: "Flamingo",

category: "birds",

img: "images/Flamingo.jpeg",

info: [

"Preserve wetlands that serve as key breeding and feeding grounds.",

"Avoid excessive tourist disturbance in flamingo colonies especially during nesting season.",

"Prevent pollution and salinity changes in wetlands caused by human development."

]

},

{

name: "Penguin",

category: "birds",

img: "images/Penguin.jpeg",

info: [

"Preserve polar ice to ensure access to breeding areas and food sources.",

"Ban oil drilling and shipping near penguin habitats to avoid spills and noise pollution.",

"Regulate overfishing of krill and small fish that penguins depend on."

]

},

{

name: "Kakapo",

category: "birds",

img: "images/Kakapo.jpeg",

info: [

"Maintain predator-free island sanctuaries where kakapos can safely breed.",

"Use tracking collars and feeding stations to monitor health and behavior.",

"Promote captive breeding programs to restore wild population."

]

},

{

name: "Shoebill",

category: "birds",

img: "images/Shoebill.jpeg",

info: [

"Protect African swamps and wetlands from encroachment and drainage.",

"Prevent illegal pet trade through awareness and regulation.",

"Preserve fish populations which are vital to the shoebill’s diet."

]

},

{

name: "Hornbill",

category: "birds",

img: "images/Hornbill.jpeg",

info: [

"Ban deforestation of tropical forests that hornbills nest in.",

"Install artificial nest boxes in fragmented forest areas.",

"Educate communities on hornbill’s role in seed dispersal."

]

},

{

name: "Galápagos Tortoise",

category: "reptiles",

img: "images/Galapagos Tortoise.jpeg",

info: [

"Control invasive species on islands that prey on tortoise eggs.",

"Restore native vegetation to provide food and shelter.",

"Support breeding and release programs for population growth."

]

},

{

name: "Leatherback Turtle",

category: "reptiles",

img: "images/Leatherback Turtle.jpeg",

info: [

"Ban beach vehicle traffic that crushes nests and hatchlings.",

"Enforce bycatch regulations to avoid turtle entanglement in fishing gear.",

"Monitor nesting beaches for protection and data collection."

]

},

{

name: "Gila Monster",

category: "reptiles",

img: "images/Gila Monster.jpeg",

info: [

"Preserve desert habitats from urban sprawl and mining.",

"Educate public about the species' low threat and ecological role.",

"Ban illegal pet trade of venomous reptiles."

]

},

{

name: "Green Sea Turtle",

category: "reptiles",

img: "images/Green Sea Turtle.jpeg",

info: [

"Ban plastic waste which turtles often ingest, mistaking it for jellyfish.",

"Protect nesting beaches through volunteer patrols and conservation zoning.",

"Support marine conservation groups working on turtle tagging and habitat restoration."

]

},

{

name: "Komodo Dragon",

category: "reptiles",

img: "images/Komodo Dragon.jpeg",

info: [

"Protect native island habitats from encroachment and illegal developments.",

"Ban illegal hunting and regulate tourism to prevent habitat degradation.",

"Control tourism impact by establishing limits and providing eco-guided experiences."

]

},

{

name: "King Cobra",

category: "reptiles",

img: "images/King Cobra.jpeg",

info: [

"Educate rural communities on snake conservation and safe co-existence.",

"Preserve forest cover to provide adequate shelter and prey for cobras.",

"Avoid conflict zones by relocating snakes safely from human dwellings."

]

},

{

name: "Chameleon",

category: "reptiles",

img: "images/Chameleon.jpeg",

info: [

"Avoid habitat loss due to urban sprawl and deforestation in tropical zones.",

"Ban pet trade of wild chameleons which harms local populations.",

"Maintain biodiversity by protecting forests with diverse flora and insect life."

]

},

{

name: "Gharial",

category: "reptiles",

img: "images/Gharial.jpeg",

info: [

"Restore river flows that are essential for egg laying and juvenile development.",

"Prevent sand mining along riverbanks where gharials nest.",

"Protect breeding grounds with fences and awareness programs in rural communities."

]

},

{

name: "Blue Whale",

category: "aquatics",

img: "images/Blue Whale.jpeg",

info: [

"Ban ship strikes by designating marine traffic-free zones in whale-rich areas.",

"Limit ocean noise from ships and drilling which interferes with whale communication.",

"Protect krill sources by controlling commercial fishing and preserving ocean health."

]

},

{

name: "Dolphin",

category: "aquatics",

img: "images/Dolphin.jpeg",

info: [

"Avoid fishing nets by using dolphin-safe fishing techniques and gear.",

"Preserve coral reefs which are vital ecosystems for many dolphin species.",

"Stop pollution from plastics and chemicals that enter the marine food chain."

]

},

{

name: "Clownfish",

category: "aquatics",

img: "images/Clownfish.jpeg",

info: [

"Save coral reefs that clownfish depend on for shelter and breeding.",

"Reduce aquarium trade by promoting captive-bred fish and regulating wild collection.",

"Promote marine parks where coral ecosystems are protected from tourism and fishing."

]

},

{

name: "Great White Shark",

category: "aquatics",

img: "images/Great White Shark.jpeg",

info: [

"Prevent shark finning by enforcing fishing bans and raising awareness globally.",

"Respect ocean zones by limiting human activity in known shark territories.",

"Support research tagging to track migration and inform conservation policy."

]

},

{

name: "Jellyfish",

category: "aquatics",

img: "images/Jellyfish.jpeg",

info: [

"Maintain plankton population that jellyfish feed on through clean water initiatives.",

"Monitor water temperature changes caused by climate change and industrial activity.",

"Protect from pollutants that damage delicate marine food webs and disrupt balance."

]

},

{

name: "Manatee",

category: "aquatics",

img: "images/Manatee.jpeg",

info: [

"Install speed restrictions for boats in manatee zones.",

"Preserve seagrass beds and freshwater sources they depend on.",

"Rescue and rehabilitate injured or orphaned manatees."

]

},

{

name: "Seahorse",

category: "aquatics",

img: "images/Seahorse.jpeg",

info: [

"Regulate traditional medicine and curio trade affecting seahorse populations.",

"Protect coral reefs and seagrass beds from degradation.",

"Ban destructive fishing practices like bottom trawling in key habitats."

]

},

{

name: "Orca (Killer Whale)",

category: "aquatics",

img: "images/Orca.jpeg",

info: [

"Ban captivity and promote wild research instead of entertainment use.",

"Protect prey species like salmon from overfishing and damming.",

"Limit noise pollution in oceans that disrupts whale communication."

]

},

{

name: "Atlas Moth",

category: "insects",

img: "images/Atlas Moth.jpeg",

info: [

"Preserve tropical rainforests where this species thrives.",

"Ban collection and illegal trade of rare moths.",

"Promote biodiversity conservation that includes less-known insect species."

]

},

{

name: "Praying Mantis",

category: "insects",

img: "images/Praying Mantis.jpeg",

info: [

"Avoid pesticide use that kills beneficial predators like mantises.",

"Encourage planting of gardens and crops that support mantis habitats.",

"Educate children on the importance of predator insects in ecosystems."

]

},

{

name: "Bumblebee",

category: "insects",

img: "images/Bumblebee.jpeg",

info: [

"Plant native flowering plants that bloom across seasons.",

"Promote pesticide-free agriculture and landscaping.",

"Support citizen science efforts to track pollinator populations."

]

},

{

name: "Honeybee",

category: "insects",

img: "images/Honeybee.jpeg",

info: [

"Avoid pesticide use which weakens bee immune systems and reduces colony survival.",

"Plant wildflowers and native plants to support nectar and pollen supply.",

"Support local beekeepers who play a crucial role in pollination and food security."

]

},

{

name: "Monarch Butterfly",

category: "insects",

img: "images/Monarch Butterfly.jpeg",

info: [

"Grow milkweed, the only plant monarch caterpillars feed on.",

"Reduce deforestation in their migration paths, especially in Mexico and the U.S.",

"Protect migratory routes by creating butterfly-friendly habitats along highways."

]

},

{

name: "Dragonfly",

category: "insects",

img: "images/Dragonfly.jpeg",

info: [

"Protect freshwater habitats where dragonflies breed and lay eggs.",

"Limit pesticide run-off into lakes and rivers which affects aquatic insect larvae.",

"Conserve wetlands that provide ideal conditions for dragonfly life cycles."

]

},

{

name: "Ladybug",

category: "insects",

img: "images/Ladybug.jpeg",

info: [

"Limit chemical use that kills natural aphid predators like ladybugs.",

"Encourage garden biodiversity to provide a variety of prey and plants.",

"Support organic farming practices that preserve beneficial insect populations."

]

},

{

name: "Firefly",

category: "insects",

img: "images/Firefly.jpeg",

info: [

"Reduce light pollution which disrupts mating signals used by fireflies.",

"Preserve moist habitats such as marshes and streams where fireflies lay eggs.",

"Avoid urban sprawl which encroaches on natural firefly habitats."

]

}

];

const modal = document.getElementById('modal');

const closeModalBtn = document.getElementById('close-modal');

const speciesTitle = document.getElementById('species-title');

const speciesImg = document.getElementById('species-img');

const speciesInfo = document.getElementById('species-info');

// Render species cards

speciesList.forEach(species => {

const card = document.createElement('div');

card.className = 'species-card';

card.innerHTML = `

<img class="species-image" src="${species.img}" alt="${species.name}">

<div class="species-info">

<h3 class="species-name">${species.name}</h3>

<span class="species-status">Endangered</span>

</div>

`;

card.addEventListener('click', () => openModal(species));

document.getElementById(species.category).appendChild(card);

});

function openModal(species) {

speciesTitle.textContent = species.name;

speciesImg.src = species.img;

speciesImg.alt = species.name;

speciesInfo.innerHTML = species.info

.map(item => `<li>${item}</li>`)

.join('');

modal.classList.add('active');

document.body.style.overflow = 'hidden';

}

function closeModal() {

modal.classList.remove('active');

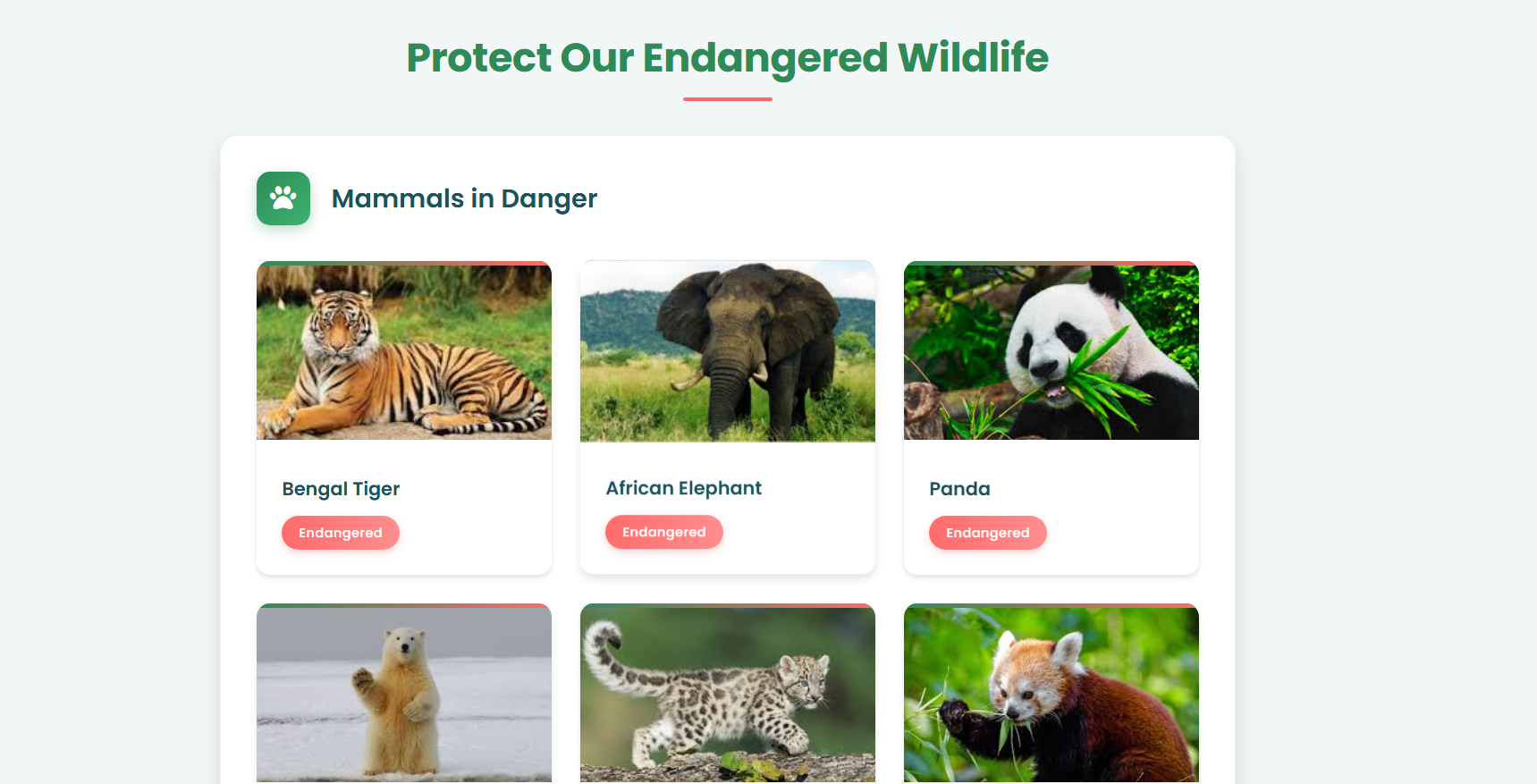
document.body.style.overflow = 'auto';

}

// Event Listeners

closeModalBtn.addEventListener('click', closeModal);

window.addEventListener('click', (e) => e.target === modal && closeModal());

document.addEventListener('keydown', (e) => e.key === 'Escape' && closeModal());  
  
  
OUTPUT DESIGN  
  


## ****CONCLUSION AND SCOPE FOR FUTURE ENHANCEMENT****

### ****Conclusion:****

The **Zoo Management System** is a static, web-based educational platform built with **HTML, CSS, and JavaScript**, hosted and integrated using **AWS services**. Its purpose is to spread awareness about endangered wildlife through an engaging, categorized layout, allowing users to explore species interactively via modal popups.

The project utilizes modern web technologies and offers:

* **Responsive design** for mobile, tablet, and desktop
* **Lightweight deployment** on the cloud
* **Interactive UI** using only front-end technologies
* **AWS-powered storage and access**

Hosting the site with **AWS S3**, managing user permissions through **IAM**, and ensuring secure access shows that the system is not only educational but also built on **scalable cloud infrastructure**.

### ****Scope for Future Enhancement (AWS-Based and Static-Friendly):****

Here are realistic and scalable enhancement suggestions aligned with your AWS setup:

#### **1. AWS S3 + JSON-Based Content Management**

* Store species details (name, category, description, image URLs) in a **JSON file on S3**.
* Use JavaScript to load and render data dynamically.
* No need to update HTML manually—just update the JSON on S3.

#### **2. AWS IAM Access Control**

* Use **IAM roles and policies** to restrict who can upload or modify species data in the S3 bucket.
* Ideal for maintaining a secure and controlled environment if multiple people manage content.

#### **3. CloudFront for Global Performance**

* Distribute your site using **AWS CloudFront CDN** to improve speed and access globally.
* Helps with faster load times, especially for media-rich content (images, sounds).

#### **4. AWS Amplify (Optional)**

* Use **AWS Amplify** for easier CI/CD, file hosting, and optional integrations like:
  + User authentication (login/registration for volunteers)
  + Serverless APIs (e.g., feedback form or conservation tips)
  + Automatic deployment from GitHub

#### **5. AWS CloudWatch or Amazon QuickSight**

* Track visitor behavior or website usage (like most clicked animals) via **CloudWatch logs** or integrate with **QuickSight** for visual dashboards.
* Useful for improving educational impact and measuring user engagement.

#### **6. Progressive Web App (PWA) with AWS Hosting**

* Convert the site into a **Progressive Web App** (PWA) so users can “install” it on their devices.
* All hosted on S3 + CloudFront, with offline access via service workers.

#### **7. Multimedia and Educational Enhancements**

* Store and stream **educational videos, animal sounds, and voiceovers** from S3.
* Embed or play them dynamically on modals to boost interactivity.

#### **­­8. Multi-language Support via Static JSON**

* Upload localized content files (e.g., species\_en.json, species\_es.json) on S3.
* Let users switch language dynamically through JavaScript, all without a backend.

## ****BIBLIOGRAPHY AND REFERENCE****

### ****Web Development & Design Resources:****

1. MDN Web Docs – <https://developer.mozilla.org>  
   Comprehensive documentation on HTML, CSS, and JavaScript.
2. W3Schools – <https://www.w3schools.com>  
   Basic tutorials and code examples for front-end web technologies.
3. FontAwesome – <https://fontawesome.com>  
   Used for integrating icons into the website.
4. Google Fonts – <https://fonts.google.com>  
   Source for custom web fonts to enhance UI design.
5. Stack Overflow – <https://stackoverflow.com>  
   Community-based support and troubleshooting for development issues.

### ****AWS Documentation & Services:****

1. AWS S3 Documentation – <https://docs.aws.amazon.com/s3>  
   Used for static website hosting and media file storage.
2. AWS IAM Documentation – <https://docs.aws.amazon.com/iam>  
   Guidance on securing access to AWS resources.
3. AWS CloudFront – <https://docs.aws.amazon.com/cloudfront>  
   Used for content delivery and improved performance (optional).

### ****Wildlife & Conservation References:****

1. World Wildlife Fund (WWF) – <https://www.worldwildlife.org>  
   Information on endangered species and conservation efforts.
2. IUCN Red List – <https://www.iucnredlist.org>  
   Reliable source of global conservation status of animal species.
3. National Geographic – <https://www.nationalgeographic.com/animals>  
   Used for species facts, images, and awareness content.

### ****Design Inspiration:****

1. Dribbble – <https://dribbble.com>  
   UI/UX design ideas and modern layout inspirations.
2. CodePen – <https://codepen.io>  
   Front-end examples for responsive cards, modals, and animations.