

Sancturia Wildlife - Complete Project Documentation

Project Overview

Sancturia Wildlife is a web-based platform dedicated to wildlife conservation in India. The platform connects compassionate donors with wildlife sanctuaries across India, enabling direct financial support for conservation efforts. Users can explore sanctuaries, make donations, and symbolically adopt animals to support their care.

Project Structure

Sancturia-Wildlife/		
—	index.html	# Homepage
—	homeStyle.css	# Homepage styles
—	home.js	# Homepage JavaScript
—	Assets_TBU/	
—	Background Images/	# Background images & logos
—	Animal Images/	# Sanctuary and animal images
—	Logos Icons/	# Icons and animations
—	pages/	
—	about/	
—	about.html	# About page
—	aboutStyle.css	# About page styles
—	donate/	
—	donate.html	# Donation page
—	donateStyle.css	# Donation styles
—	donate.js	# Donation JavaScript
—	thankyou.html	# Thank you page
—	sanctuaries/	
—	sanctuaries.html	# Sanctuaries listing
—	sanctuariesStyle.css	# Sanctuaries styles
—	sanctuaries.js	# Sanctuaries JavaScript
—	adopt/	
—	adopt.html	# Adoption page (under construction)
—	login-signup/	
—	login.html	# Login page
—	signup.html	# Signup page
—	signup.php	# Signup backend
—	signupStyle.css	# Login/Signup styles
—	dashboard/	
—	dashboard.php	# User dashboard
—	navbar/	

Technology Stack

Frontend

- **HTML5** - Structure and content
- **CSS3** - Styling and animations
- **JavaScript (ES6+)** - Client-side interactivity
- **Bootstrap 5.0.2** - Responsive framework
- **Google Fonts** - Sofia font family

Backend

- **PHP** - Server-side processing
- **Firebase Authentication** - User authentication
- **Firebase Firestore** - Database storage
- **Firebase Realtime Database** - User data storage

External Libraries

- Bootstrap 5.0.2 (CSS & JS)
 - Google Fonts (Sofia family)
-

Core Features

1. Homepage (index.html)

Purpose: Landing page showcasing the mission and featured sanctuaries

Key Elements:

- Hero section with tagline: "Protecting Indian Wildlife Sanctuaries"
- Quick donation section with preset amounts (₹100, ₹500, ₹1000, ₹2000)
- Custom donation input field

- Recurring donation options (monthly/yearly)
- Featured sanctuary cards with images (Jim Corbett, Ranthambore, Chilika, Nal Sarovar)
- Navigation to sanctuary details via "Visit" buttons
- Jungle-themed background image

JavaScript Functionality (home.js):

```
javascript

// Preset donation buttons set placeholder values
btnRs.forEach(btn => {
  btn.addEventListener('click', (e) => {
    donationPreset = e.target.innerText;
    localStorage.setItem('DonationPreset', donationPreset);
    donationAmt.placeholder = btn.innerText;
  });
});

// Sanctuary redirection stores sanctuary name in localStorage
parkBtns.forEach(btn => {
  btn.addEventListener('click', (e) => {
    sancName = e.target.name;
    localStorage.setItem('SanctuaryName', sancName);
    window.location.href = './sancturies.html';
  });
});

// Radio button toggle functionality
radioButtons.forEach(radio => {
  radio.addEventListener('click', function() {
    if (this === lastChecked) {
      this.checked = false;
      lastChecked = null;
    } else {
      lastChecked = this;
    }
  });
});
```

CSS Highlights (homeStyle.css):

- Full viewport jungle background with blur effect

- Glassmorphism donation section ((backdrop-filter: blur(1px)))
 - Custom golden-themed radio buttons
 - Responsive design with breakpoints at 1199px, 991px, 872px, 768px, 576px
 - Hover animations on sanctuary images
-

2. Sanctuaries Page (sancturies.html)

Purpose: Comprehensive listing of 52 Indian wildlife sanctuaries with search functionality

Key Features:

- Search bar for finding specific sanctuaries
- Dynamic grid layout (auto-fit, minmax(270px, 1fr))
- Modal popup for searched sanctuary details
- Shuffled sanctuary display on page load
- Direct links to official sanctuary websites

Sanctuary Data Structure (sancturies.js):

javascript

```
class Sanctuary {  
  constructor(id, name, location, description, img, website) {  
    this.id = id;  
    this.name = name;  
    this.location = location;  
    this.description = description;  
    this.img = img;  
    this.website = website;  
  }  
  
  getInfo() {  
    return `${this.id}, ${this.name}, ${this.location}, ${this.description}, ${this.img}, ${this.website}`;  
  }  
}  
  
// 52 sanctuaries including:  
// - Ranthambore National Park (Rajasthan)  
// - Kaziranga National Park (Assam)  
// - Jim Corbett National Park (Uttarakhand)  
// - Sundarban National Park (West Bengal)  
// ... and 48 more
```

Search Functionality:

javascript

```

function findSanctuaryByName(name) {
  return sanctuaries.find(sanctuary =>
    sanctuary.name.split(' ')[0].toLowerCase() === name.split(' ')[0].toLowerCase()
  );
}

function searchSanc() {
  try {
    searchedModal.style.display = 'block';
    searchedModal.innerHTML = null;
    searchVal = searchBar.value;
    searchInfo = findSanctuaryByName(searchVal).getInfo();
    searchInfo = searchInfo.split(', ');

    // Create modal card with sanctuary details
    const searchedCard = `
      <div class="sanctuary-card">
        
        <h3>${searchInfo[1]}</h3>
        <p>Location: ${searchInfo[2]}</p>
        <p>${searchInfo[3]}</p>
        <button><a href="${searchInfo[5]}" target="_blank">Get Info Here</a></button>
        <a class="closeBtn btn btn-danger">Close</a>
      </div>
    `;

    searchedModal.innerHTML += searchedCard;
    searchedModal.animate(modalAnim, modalAnimTime);
  } catch (error) {
    alert("No Such Sanctuary in List!");
  }
}

```

Integration with Homepage:

- If redirected from homepage, automatically searches for sanctuary stored in localStorage
- Smooth scroll to top after search

3. Donation Page (donate.html)

Purpose: Tiered donation system with donor information collection

Donation Tiers:

1. **Tier Soil** - ₹5001+ (Leaves icon with animation)
2. **Tier Habitat** - ₹7501+ (Elephant icon with animation)
3. **Tier Earth** - ₹10001+ (Earth icon with animation)

Form Fields:

- Name (text, required)
- Email (email, required)
- Phone Number (tel, 10 digits, required)
- Donation Amount (number, required)
- Recurring options (monthly/yearly radio buttons)

JavaScript Interactions (donate.js):

```
javascript

// Preset donation from homepage
let donationValue = localStorage.getItem('DonationPreset');
if(donationValue !== null){
  donationPlaceholder.placeholder = donationValue;
}

// Animated tier card hover effects
tCard1.addEventListener('mouseover', ()=>{
  tVid1.loop = true;
  tVid1.play()
  tVid1.style.scale = '120%';
});

// Donation button animation
donateBtn.addEventListener('click', ()=>{
  tierIconMove() // Triggers all tier animations simultaneously
});
```

Design Features:

- Wildlife background video (wildlifeVideo.mp4)
- Glassmorphism form design

- Golden border accents
 - Responsive grid layout
 - Footer with social media links
-


4. About Page (about.html)

Purpose: Organization information and mission statement

Sections:

1. **Our Mission** - Platform purpose and conservation goals
2. **Our Vision** - Long-term ecological objectives
3. **Our Story** - Founded in 2024, addressing sanctuary resource limitations
4. **How We Work** - User-friendly donation process explanation
5. **Join Us** - Call-to-action for supporter engagement

Design:

- Header with green ( #4CAF50) accent color
 - Content in centered card (max-width: 70vw)
 - Warm background (rgba(255, 244, 221, 0.897))
 - HR separators between sections
 - Responsive down to 600px
-

5. Authentication System

Signup Page (signup.html)

Form Fields:

- Full Name
- Email Address
- Password
- Confirm Password
- Terms of Service & Privacy Policy agreement

Backend Processing (signup.php):

```
php

// Firebase REST API integration
$firebase_api_key = "AIzaSyDsYiL3F1F6hcTHx3x7v9yRJMBEL-9iKGc";
$url = "https://identitytoolkit.googleapis.com/v1/accounts:signUp?key=" . $firebase_api_key;

// Create user account
$postData = [
    "email" => $email,
    "password" => $password,
    "returnSecureToken" => true
];

// Store additional user data in Firebase Realtime Database
$database_url = "https://sancturia-wildlife-default-rtdb.firebaseio.com/users/$firebase_uid.json";
$extraData = [
    "name" => $name,
    "email" => $email,
    "created_at" => date("Y-m-d H:i:s")
];



// Redirect to dashboard on success
header("Location: /pages/dashboard/dashboard.php");
```

Login Page (login.html)

Form Fields:

- Email
- Password
- "Don't have an account?" link to signup

Design Features:

- Background image with overlay gradient
- Glassmorphism card design
- Slide-down entrance animation
- Green ( #114357) and coral ( #f29492) color scheme

6. User Dashboard (dashboard.php)

Purpose: Personalized user control panel showing activity and recommendations


Authentication Check:

```
php


session_start();
if (!isset($_SESSION['uid']) || !isset($_SESSION['idToken'])) {
    header("Location: login.php");
    exit();
}
```

Dashboard Sections:


1. User Profile Card

- Name, Email, UID
- Signup date
- Green header ( #4CAF50)

2. Donations Summary

- Total donated amount (₹)
- Recent donations list (last 5)
- Individual donation details (amount, sanctuary, timestamp)
- Warning ( #FFC107) header

3. Adoptions Summary

- Total adoptions count
- Recent adoptions list
- Animal details (name, type, timestamp)
- Info ( #17A2B8) header

4. Recommended Sanctuaries

- 3 featured sanctuaries
- Name, location, description
- "Explore" action button

Firestore Integration:

php

```
$projectId = "sancturia-wildlife";  
$firestoreUrl = "https://firestore.googleapis.com/v1/projects/$projectId/databases/(default)/documents";  
  
// Fetch user profile  
$userDoc = fetchFromFirestore("$firestoreUrl/users/$uid");  
  
// Query donations by user ID  
$donationsQuery = "$firestoreUrl/donations?where=uid%3D'$uid'&orderBy=timestamp%20desc&limit=5";  
  
// Query adoptions by user ID  
$adoptionsQuery = "$firestoreUrl/adoptions?where=uid%3D'$uid'&orderBy=timestamp%20desc";
```



7. Adoption Page (adopt.html)

Status: Under Construction

Planned Features:

- Symbolic animal adoption
- Personalized animal naming
- Direct contribution to animal care
- Support for sanctuary wildlife

Design:

- Construction icon ()
- Yellow ( #FFCC00) theme
- Informational message about upcoming feature
- "Back to Home" CTA button

8. Thank You Page (thankyou.html)

Purpose: Donation confirmation and gratitude message

Features:

- Centered thank you message
- Pulsing animation effect
- Background image with multiply blend mode
- "Go Back to Home" button (orange accent)
- Gradient background (green to blue)

CSS Animation:

css

```
@keyframes pulse {  
  0%{ transform: scale(1); }  
  50%{ transform: scale(1.05); }  
  100%{ transform: scale(1); }  
}
```

9. Reusable Navbar Component

Implementation (navbar.html + navbar.js):

javascript

```
// Load navbar dynamically  
document.addEventListener("DOMContentLoaded", function () {  
  fetch("/pages/navbar/navbar.html")  
    .then(response => response.text())  
    .then(data => {  
      document.getElementById("navbar-placeholder").innerHTML = data;  
  
      // Highlight active page  
      const currentPage = window.location.pathname.split("/").pop();  
      document.querySelectorAll("#navBarList a").forEach(link => {  
        const linkPage = link.getAttribute("href").split("/").pop();  
        if(linkPage === currentPage) {  
          link.classList.remove("btn-outline-dark", "btn-outline-light");  
          link.classList.add("btn-dark");  
        }  
      });  
    });  
});
```

Navbar Links:

- Home
- About
- Donate
- Sanctuaries
- Adopt
- Sign Up / Login buttons

Responsive Features:

- Collapsible menu for mobile (navbar-toggler)
 - Bootstrap responsive classes
 - Active page highlighting
-

Data Flow & User Journey

1. First-Time Visitor Journey

Homepage → Explore Sanctuaries → Select Sanctuary → View Details →
Donate → Fill Form → Thank You Page

2. Returning User Journey

Login → Dashboard → View Activity → Donate Again / Explore New Sanctuaries

3. Donation Flow

Homepage (preset amount selection) → localStorage storage →
Donate Page (form autofill) → Submit → Thank You Page

4. Sanctuary Exploration Flow

Homepage (sanctuary image click) → localStorage storage →
Sanctuaries Page (auto-search) → Modal popup → Official website

LocalStorage Usage

The project uses browser localStorage for cross-page data persistence:

1. **DonationPreset** - Stores selected donation amount
2. **SanctuaryName** - Stores selected sanctuary for search

Example:

```
javascript

// Set on homepage
localStorage.setItem('DonationPreset', '1000');

// Retrieve on donate page
let donationValue = localStorage.getItem('DonationPreset');
donationPlaceholder.placeholder = donationValue;

// Clear storage
localStorage.clear();
```

Firestore Configuration

Authentication

- **Project ID:** sancturia-wildlife
- **API Key:** AIzaSyDsYiL3F1F6hcTHx3x7v9yRJMBEL-9iKGc
- **Authentication Endpoint:** identitytoolkit.googleapis.com/v1/accounts:signUp

Realtime Database

- **URL:** <https://sancturia-wildlife-default-rtdb.firebaseio.com>
- **Structure:**

```
users/  
  {uid}/  
    - name  
    - email  
    - created_at  
    - donation_total  
    - adoptions
```

Firestore

- **Collections:**
 - `users` - User profiles
 - `donations` - Donation records
 - `adoptions` - Animal adoption records
 - `sanctuaries` - Sanctuary information
-

Design System

Color Palette

- **Primary Green:** `#4CAF50` (buttons, headers)
- **Golden/Warning:** `#FFA500`, `#FFD700`, goldenrod (accents, donations)
- **Dark:** `#333`, `#114357` (text, borders)
- **Light:** `#f4f4f4`, aliceblue (backgrounds)
- **Coral:** `#f29492` (gradients)

Typography

- **Primary Font:** Sofia (Google Fonts)
- **Fallback:** Arial, sans-serif
- **Hero Text:** 60px (desktop)
- **Headings:** 30-50px
- **Body:** 18-22px

Spacing & Layout

- **Container Max Width:** 70-80vw
- **Border Radius:** 0.8rem - 1.5rem
- **Card Padding:** 20-40px
- **Grid Gap:** 1.5rem

Effects

- **Backdrop Filter:** blur(1-3px) for glassmorphism
 - **Box Shadow:** 0 2px 10-40px rgba(0, 0, 0, 0.1-0.4)
 - **Transitions:** 0.3-0.5s ease
 - **Hover Scale:** 101-105%
-

Responsive Breakpoints

CSS

```
@media (min-width: 1550px) /* Large desktops */
@media (max-width: 1199px) /* Hide column1, column4 */
@media (max-width: 991px) /* Adjust hero text, navbar */
@media (max-width: 872px) /* Donation section width */
@media (max-width: 768px) /* Hide all sanctuary columns */
@media (max-width: 576px) /* Full width donation section */
@media (max-width: 600px) /* About page adjustments */
```

Key JavaScript Patterns

1. Event Delegation

javascript

```
btnRs.forEach(btn => {
  btn.addEventListener('click', (e) => {
    // Handle individual button clicks
  });
});
```


2. Array Manipulation

javascript

```
function shuffleArray(array) {  
  for (let i = 0; i < array.length-1; i++) {  
    const j = Math.floor(Math.random() * (50));  
    [array[i], array[j]] = [array[j], array[i]];  
  }  
  return array;  
}
```

3. Dynamic HTML Generation

javascript

```
sanctuaries.forEach(sanctuary => {  
  const sanctuaryCard = `  
    <div class="sanctuary-card">  
        
      <h3>${sanctuary.name}</h3>  
      <p>${sanctuary.description}</p>  
    </div>  
  `;  
  container.innerHTML += sanctuaryCard;  
});
```

4. Web Animations API

javascript

```
const modalAnim = [  
  {opacity: 0, transform: 'scale(80%)'},  
  {opacity: 1, transform: 'scale(100%)'}  
];  
searchedModal.animate(modalAnim, {duration: 500, easing: 'ease'});
```

Security Considerations

Current Implementation

1. **Firebase Authentication** - Handles password hashing

2. **PHP Session Management** - Server-side authentication
3. **Form Validation** - Client-side and server-side
4. **HTTPS Required** - Firebase endpoints use HTTPS

Recommended Improvements

1. **Environment Variables** - Move API keys to .env file
 2. **CSRF Protection** - Add tokens to forms
 3. **Input Sanitization** - Enhanced XSS prevention
 4. **Rate Limiting** - Prevent abuse on authentication endpoints
 5. **Password Strength** - Enforce complexity requirements
-

Browser Compatibility

Supported Features

- **CSS Grid** - Modern layout system
- **Flexbox** - Responsive alignment
- **localStorage** - Data persistence
- **fetch API** - AJAX requests
- **ES6+ JavaScript** - Arrow functions, destructuring, template literals
- **Web Animations API** - Smooth transitions

Minimum Requirements

- Chrome 60+
 - Firefox 55+
 - Safari 11+
 - Edge 79+
-

Performance Optimization

Current Optimizations

1. **Image Loading** - Background images as fixed elements
2. **CSS Transitions** - Hardware-accelerated transforms
3. **Event Handling** - Efficient event delegation
4. **Code Splitting** - Separate CSS/JS files per page

Recommended Improvements

1. **Image Optimization** - Compress and use WebP format
 2. **Lazy Loading** - Defer offscreen images
 3. **Minification** - Minify CSS/JS files
 4. **CDN Usage** - Serve static assets from CDN
 5. **Caching** - Implement browser caching headers
-

Future Enhancements

Planned Features

1. **Animal Adoption System** - Complete adopt.html functionality
 2. **Payment Gateway Integration** - Razorpay/Stripe for donations
 3. **Email Notifications** - Donation receipts and updates
 4. **Social Sharing** - Share sanctuaries on social media
 5. **Admin Dashboard** - Manage sanctuaries and donations
 6. **Multilingual Support** - Hindi and regional languages
 7. **Mobile App** - Native iOS/Android applications
 8. **Volunteer Registration** - On-site volunteer opportunities
 9. **Impact Reports** - Show donation utilization
 10. **Leaderboards** - Top donors and supporters
-

Complete File Reference

HTML Files (7)

1. **index.html** - Homepage with hero and donation preview
2. **sancturies.html** - Full sanctuary listing with search
3. **donate.html** - Tiered donation page with form
4. **about.html** - Organization information
5. **login.html** - User login form
6. **signup.html** - User registration form
7. **adopt.html** - Under construction page
8. **thankyou.html** - Donation confirmation
9. **navbar.html** - Reusable navigation component

CSS Files (5)

1. **homeStyle.css** - Homepage styles (320 lines)
2. **sancturiesStyle.css** - Sanctuary page styles
3. **donateStyle.css** - Donation page styles
4. **aboutStyle.css** - About page styles
5. **signupStyle.css** - Authentication page styles

JavaScript Files (4)

1. **home.js** - Homepage interactions (~80 lines)
2. **sancturies.js** - Sanctuary management (~200 lines)
3. **donate.js** - Donation interactions (~50 lines)
4. **navbar.js** - Dynamic navbar loader (~20 lines)

PHP Files (2)

1. **signup.php** - User registration backend (~80 lines)
 2. **dashboard.php** - User dashboard with Firestore (~120 lines)
-

Database Schema

Firestore Collections

users/

```
json
{
  "uid": "unique_user_id",
  "fields": {
    "name": {"stringValue": "John Doe"},
    "email": {"stringValue": "john@example.com"},
    "donation_total": {"integerValue": 5000},
    "adoptions": {"integerValue": 2},
    "createdAt": {"timestampValue": "2024-01-01T00:00:00Z"}
  }
}
```

donations/

```
json
{
  "donation_id": "unique_donation_id",
  "fields": {
    "uid": {"stringValue": "user_id"},
    "amount": {"integerValue": 1000},
    "sanctuary": {"stringValue": "Ranthambore National Park"},
    "recurring": {"stringValue": "monthly"},
    "timestamp": {"timestampValue": "2024-01-15T10:30:00Z"}
  }
}
```

adoptions/

```
json
```

```
{
  "adoption_id": "unique_adoption_id",
  "fields": {
    "uid": {"stringValue": "user_id"},
    "animal_name": {"stringValue": "Shera"},
    "type": {"stringValue": "Tiger"},
    "sanctuary": {"stringValue": "Jim Corbett"},
    "timestamp": {"timestampValue": "2024-01-20T14:00:00Z"}
  }
}
```

sanctuaries/

```
json
{
  "sanctuary_id": "unique_sanctuary_id",
  "fields": {
    "name": {"stringValue": "Ranthambore National Park"},
    "location": {"stringValue": "Rajasthan"},
    "description": {"stringValue": "Famous for Bengal tigers..."},
    "website": {"stringValue": "https://www.ranthamborenationalpark.com/"},
    "image": {"stringValue": "/Assets_TBU/Animal Images/img1.jpg"}
  }
}
```

API Endpoints

Firestore Authentication

```
POST https://identitytoolkit.googleapis.com/v1/accounts:signUp
POST https://identitytoolkit.googleapis.com/v1/accounts:signInWithPassword
POST https://identitytoolkit.googleapis.com/v1/accounts:sendOobCode
```

Firestore

```
GET /v1/projects/{projectId}/databases/(default)/documents/users/{uid}
GET /v1/projects/{projectId}/databases/(default)/documents/donations
POST /v1/projects/{projectId}/databases/(default)/documents/donations
GET /v1/projects/{projectId}/databases/(default)/documents/sanctuaries
```

Firestore Realtime Database

```
PUT  /{database}/users/{uid}.json
```

```
GET  /{database}/users/{uid}.json
```

Error Handling

Client-Side

```
javascript
```

```
try {
  searchInfo = findSanctuaryByName(searchVal).getInfo();
  // Display results
} catch (error) {
  searchedModal.style.display = 'none';
  alert("No Such Sanctuary in List!");
}
```

Server-Side (PHP)

```
php
```

```
if (isset($data["error"])) {
  echo "❌ Error: " . $data["error"]["message"];
} else {
  // Success handling
  header("Location: /pages/dashboard/dashboard.php");
  exit();
}
```

Deployment Checklist

- ☐ Update Firebase API keys to production values
- ☐ Configure HTTPS for all pages
- ☐ Set up proper .htaccess rules
- ☐ Optimize and compress images
- ☐ Minify CSS and JavaScript files
- ☐ Test all forms and validations

- ☐ Set up error logging
 - ☐ Configure backup systems
 - ☐ Test responsive design on real devices
 - ☐ Implement SEO meta tags
 - ☐ Set up Google Analytics
 - ☐ Test payment integration (when implemented)
-

Testing Scenarios

User Registration

1. Valid signup with all fields
2. Password mismatch error
3. Existing email error
4. Invalid email format

Donation Flow

1. Preset amount selection
2. Custom amount input
3. Form validation (phone number, email)
4. Recurring donation options
5. localStorage persistence across pages

Sanctuary Search

1. Search by first word (case-insensitive)
2. Search with full name
3. Non-existent sanctuary error
4. Modal open/close animations
5. Direct link to official website

Dashboard

1. User authentication check
2. Display user profile data

3. Show recent donations (limit 5)
 4. Show recent adoptions
 5. Display recommended sanctuaries (3)
-

Maintenance Guidelines

Regular Tasks

1. **Database Cleanup** - Archive old donation records
2. **Image Management** - Optimize new sanctuary images
3. **Link Verification** - Check external sanctuary websites
4. **Security Updates** - Update Bootstrap and libraries
5. **Backup Schedule** - Weekly database backups

Monitoring

1. **User Activity** - Track signups and donations
 2. **Error Logs** - Monitor PHP and JavaScript errors
 3. **Performance Metrics** - Page load times
 4. **Conversion Rates** - Donation completion rates
-

Contact & Support

Project Name: Sancturia Wildlife

Founded: 2024

Technology: HTML, CSS, JavaScript, PHP, Firebase

Purpose: Wildlife conservation through direct sanctuary support

Target: Indian wildlife sanctuaries and donors

License & Credits

Assets

- Background images stored in `/Assets_TB/Background Images/`
- Animal images stored in `/Assets_TB/Animal Images/` (img1.jpg - img52.jpg)

- Logo: Sancturia Logo (Green/White/Black variants)
- Icons: Leaves, Elephant, Earth animations

External Resources

- Bootstrap 5.0.2 (MIT License)
 - Google Fonts - Sofia family
 - Firebase services (Google)
-

Conclusion

Sancturia Wildlife is a comprehensive web platform designed to bridge the gap between wildlife conservation needs and donor support in India. The project demonstrates modern web development practices with responsive design, user authentication, dynamic content loading, and third-party API integration. With 52 sanctuaries currently listed and a tiered donation system, the platform provides a user-friendly experience for conservation-minded individuals to make a tangible impact on wildlife protection efforts across India.

The modular architecture allows for easy expansion, and the planned adoption system will further enhance user engagement. The combination of client-side JavaScript for interactivity and server-side PHP for security creates a robust solution that can scale with growing user demand and sanctuary partnerships.