S.D.M. COLLEGE OF ENGINEERING AND TECHNOLOGY,

DHARWAD – 580002

(An autonomous Institution affiliated to

Visvesvaraya Technological University, Belagavi -590 018)

Department of Information Science and Engineering

6th Semester B.E Academic Year 2023-24

Under the Guidance of

Prof.Varsha Jadhav

Minor Project-2 Report entitled

“WildLife Watch”

Minor Project-2 Course Code -21UISL605

Submitted By:

Sl. No.	NAME	USN
1.	Pradyumna P	2SD21IS033
2.	Vishal K Saklathi	2SD21IS061
3.	Pavan Kalburgi	2SD22IS402

S.D.M. COLLEGE OF ENGINEERING AND TECHNOLOGY, Dharwad – 580002

(An autonomous Institution affiliated to

Visvesvaraya Technological University, Belagavi -590018)

Department of Information Science Engineering

CERTIFICATE

*This is to certify that the project work entitled **"WildLife Watch"** is a bonified work carried out by **Mr Pradyumna P, Mr Vishal Krishnamurthi Saklathi, Mr Pavan Kalburgi bearing USN 2SD21IS033, 2SD21IS061, 2SD22IS402** in successfully completing the project for VI semester B.E Degree in **Information Science of Engineering and Technology, Autonomous Institution under Visvesvaraya Technological University, Belagavi** during the year 2023-2024.It is certified that all necessary suggestions indicated for internal assessment have been incorporated .The Minor Project-2 has been approved as it has successfully satisfied the academic requirements.*

Prof.Varsha Jadhav	**Prof.Varsha Jadhav**	**Dr. Jagdeesh Pujari**	**Dr. Ramesh L. Chakrasali**
Project Guide	Project Coordinator	HOD/ISE	Principal

	Examiner I	**Examiner II**
Signature with date		
Name		

TABLE OF CONTENTS

ABSTRACT

In the context of increasing human-wildlife interactions and pressing conservation concerns, innovative solutions for monitoring, managing, and conserving wildlife populations are urgently needed. "Wildlife Watch" is a cutting-edge web application designed to address these challenges by revolutionizing wildlife management and conservation efforts. This platform provides real-time wildlife sighting reporting, comprehensive educational resources, and timely threat alerts, equipping users with the necessary tools and knowledge to effectively protect and preserve wildlife populations and their habitats.

Current wildlife management systems often suffer from limited functionalities, fragmented data collection methods, and insufficient community involvement. Many existing platforms offer basic reporting and educational features but lack a cohesive, scalable framework necessary for addressing the complex conservation challenges of today. Wildlife Watch was developed to fill these gaps, offering a more holistic and user-centric approach to wildlife conservation. By enabling real-time sighting reports, access to extensive educational resources, and integrated threat alerts, Wildlife Watch significantly enhances the efficiency and effectiveness of conservation efforts.

Wildlife Watch not only fosters community engagement but also integrates data from various sources to empower individuals in protecting biodiversity. The platform serves as an educational tool, raising awareness about species, behaviors, and ecosystems, thereby encouraging support for conservation initiatives. Additionally, it offers recreational enjoyment and economic benefits through ecotourism, promoting conservation-minded practices. As a pivotal link between people and the natural world, Wildlife Watch cultivates stewardship and responsibility for biodiversity protection, making a substantial impact on the future of wildlife conservation.

ACKNOWLEDGEMENT

The successful realization of the Wildlife Watch project stands as a testament to the collective effort and unwavering dedication of numerous individuals and organizations. Foremost, we express our deepest gratitude to our esteemed mentors and advisors, whose invaluable guidance and expertise have been instrumental in navigating the complexities of the development process and shaping the vision of Wildlife Watch.

We extend heartfelt appreciation to our diligent team members, whose tireless efforts and collaborative spirit have been pivotal in bringing the Wildlife Watch platform to fruition. Their commitment to excellence and innovative thinking have been fundamental in overcoming challenges and pushing the boundaries of wildlife conservation and community engagement.

Special recognition is also extended to our funding sources and institutional partners, whose generous support and strategic collaboration have provided the necessary resources and infrastructure to realize the ambitious goals of the Wildlife Watch project. Their belief in the transformative potential of the platform has been a driving force behind its development and expansion.

We express sincere gratitude to the users and community members who have actively engaged with the Wildlife Watch platform, providing invaluable feedback, insights, and contributions that have shaped its evolution and enhanced its functionality. Their enthusiasm and commitment to wildlife conservation have been the cornerstone of Wildlife Watch's success.

We acknowledge the significant contributions of our project guide and project coordinator, **Prof.Varsha Jadhav**. Your guidance, support, and organizational efforts have been invaluable in ensuring the success of the Wildlife Watch project.

Finally, we extend our deepest thanks to our families, friends, and loved ones for their unwavering support, understanding, and encouragement throughout this journey. Their steadfast belief in our vision has been a constant source of inspiration and motivation.

The Wildlife Watch project stands as a testament to the power of collaboration, innovation, and community engagement, and we are profoundly grateful to everyone who has played a role in its development and success.

Team members:

Pradyumna P - 2SD21IS033

Vishal K Saklathi - 2SD21IS061

Pavan Kalburgi - 2SD22IS402

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Introduction

In an era marked by escalating human-wildlife interactions and growing conservation concerns, the need for innovative solutions to monitor, manage, and conserve wildlife populations has never been more urgent. "Wildlife Watch" emerges as a pioneering web application designed to revolutionize wildlife management and conservation efforts. By offering real-time wildlife sighting reporting, comprehensive educational resources, and timely threat alerts, Wildlife Watch equips users with the tools and knowledge necessary to protect and preserve wildlife populations and their habitats effectively.

Current wildlife management systems often fall short due to their limited functionalities, fragmented data collection methods, and lack of community involvement. Many existing platforms provide basic reporting and educational features but fail to integrate these components in a scalable, real-time framework essential for addressing today's complex conservation challenges. Wildlife Watch was developed in response to these shortcomings, aiming to create a more holistic and user-centric approach to wildlife conservation.

Wildlife Watch enables users to report sightings in real-time, access a wealth of educational resources, and receive alerts about potential threats to wildlife. By fostering community engagement and integrating data from various sources, the platform not only enhances the efficiency and effectiveness of conservation efforts but also empowers individuals to actively participate in protecting biodiversity. As we embark on this transformative journey, Wildlife Watch stands poised to make a significant impact on the future of wildlife conservation.

1.2 Purpose

Wildlife Watch serves a multifaceted purpose. Firstly, it acts as an educational tool, allowing individuals to learn about various species, their behaviors, and the ecosystems they inhabit. This firsthand experience fosters a deeper understanding and appreciation for wildlife conservation efforts. Secondly, it contributes directly to conservation by raising awareness of the importance of preserving natural habitats and the species within them. This awareness often translates into support for conservation initiatives and habitat protection measures. Additionally, wildlife watching provides recreational enjoyment and serves as a means of connecting with nature, offering a fulfilling hobby for people of all ages. Furthermore, it plays a role in scientific research, with observers often contributing valuable data to research projects through citizen science initiatives. Lastly, wildlife watching can have significant economic benefits for communities through ecotourism, providing revenue while promoting conservation-minded practices. Overall, wildlife watching serves as a vital link between people and the natural world, fostering stewardship and responsibility for the protection of biodiversity.

1.3 Scope

The scope of "Wild Life Watch" encompasses the development of a web platform and mobile app aimed at engaging communities in wildlife conservation and responsible interaction with natural habitats. The platform will enable users to report real-time wildlife sightings with multimedia content, access educational resources on conservation topics, and receive threat alerts related to habitat degradation or species endangerment. Additionally, the platform will foster community engagement through social features and collaborative projects, facilitating data collection for scientific research and conservation efforts.

The primary objectives include providing a user-friendly interface for wildlife reporting and education, promoting awareness of local conservation issues, and supporting community-driven conservation initiatives. The system will prioritize scalability and usability to accommodate a diverse user base, from casual wildlife enthusiasts to conservation professionals.

1.4 Problem Definition

The escalating human-wildlife interactions and inadequate existing wildlife management systems present significant challenges to conservation efforts, characterized by fragmented data collection, limited functionalities, and insufficient community involvement. Current platforms often lack the integration, scalability, and real-time capabilities necessary to address the complexities of modern wildlife conservation. This results in inefficient monitoring, delayed responses to threats, and missed opportunities for community engagement, ultimately undermining efforts to protect and preserve wildlife populations and their habitats. “WildLife Watch” is a solution to all above mentioned challenges.

1.5 Objectives

- To create an intuitive and accessible web platform that allows users to easily log wildlife sightings, track movement patterns, and be safe.

- To implement feature to enable real-time monitoring of wildlife movement and sightings, providing users with timely notifications and alerts.

- To Promote Public Awareness and Education by providing educational resources, articles about Wildlife conservation and endangered species preservation to raise public awareness and promote environmental stewardship and make community to involve in wildlife conversation.

- To establish a platform for users to provide feedback, suggestions, and reports of wildlife or conservation concerns, facilitating continuous improvement and responsiveness to community needs.

- Establish a real-time threat alert system to notify users of potential risks to wildlife, such as habitat destruction, poaching activities, and environmental hazards,

1.6 Methodology

“Wildlife Watch" is designed to make it easy for people to track and protect wildlife and themselves. With User Methodology

1. **Report Sightings:**

Data Entry: Users can report wildlife sightings through a user-friendly interface on the web application.

Details Capture: The reporting feature allows users to input detailed information, including the species observed, location, time, date, and any notable behavior.

Real-Time Submission: Submitted sightings are instantly uploaded to the platform's database, ensuring real-time data availability.

2. Learn Information about Wildlife:

Interactive Graphs and Maps: Users can access educational resources presented through interactive graphs and maps.

Species Information: Detailed profiles on various species, including their habitat, behavior, and conservation status, are available.

3. View Sightings:

Sightings Map: Users can view reported sightings on a map, filterable by species, date, and location.

Data Analytics: The platform provides analytical tools to observe trends and patterns in wildlife sightings, offering insights into animal behavior and population changes.

Admin Methodology

1. **Data Handling:**

Add Data: Administrators can input new data into the system, including updates on wildlife populations and habitat changes.

Update Data: Existing data can be updated to reflect new information or corrections, ensuring accuracy and relevance.

Delete Data: Administrators have the capability to remove outdated or incorrect data from the system.

2. View Sightings and Take Action:

Administrative Dashboard: Admins have access to a comprehensive dashboard where they can view all reported sightings.

Actionable Insights: The platform provides tools to analyze sightings data and identify potential threats or areas requiring intervention.

Decision-Making Support: Based on the data, administrators can coordinate conservation efforts, respond to threats, and implement policies to protect wildlife and their habitats. Here is the Flow chart of our project which outlines the features / steps involved

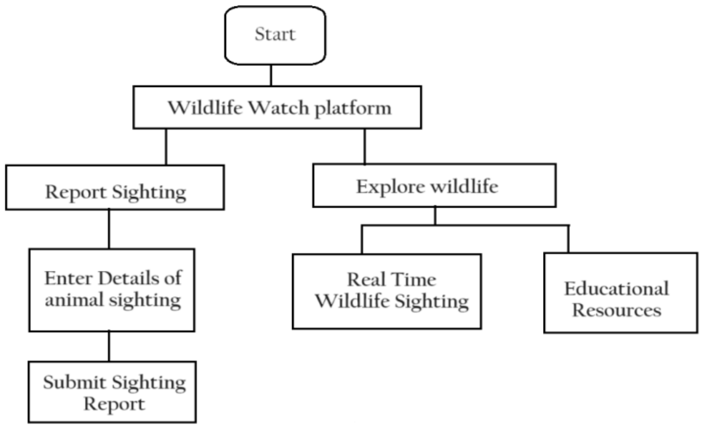


Fig 1.1 Flowchart (User)

1.7 Limitations

- Restricted coverage in remote or inaccessible areas.
- Privacy concerns related to data collection.
- Challenges in dense vegetation or rugged terrain.
- The accuracy of the data depends heavily on user reports, which can vary in precision and reliability
- Users from diverse cultural backgrounds and language proficiencies might find it challenging to use the platform effectively

CHAPTER 2

Technologies Used

Frontend Technologies:

1. HTML: HTML is used to create the basic structure and layout of the web pages.

2. CSS: CSS is used to style and design HTML elements. It allows for the separation of content and presentation, making the web pages visually appealing.

3. JavaScript: JavaScript is a programming language that enables interactive web pages. JavaScript handles functionalities like form validation, asynchronous data fetching, DOM manipulation, and interactive maps.

4.Flutter:Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications from a single codebase for any web browser. The basic component in a Flutter program is a "widget", which can in turn consist of other widgets. We have used flutter for the front-end purpose.

Backend Technologies:

1. Node.js

Node.js is a JavaScript runtime. It allows developers to run JavaScript on the server-side, enabling the creation of scalable and efficient web applications. In "WildLife Watch," Node.js is used to handle server-side operations, process requests, and interact with the database.

2. Express.js

Express.js is a web application framework for Node.js. It simplifies the development process by providing a robust set of features for routing, middleware, and handling HTTP requests and responses. In "WildLife Watch," Express.js is used to create the server, define routes for different endpoints, and manage middleware for request processing.

3. MongoDB

MongoDB is a NoSQL database known for its flexibility and scalability. It stores data in JSON-like documents, making it easy to manage hierarchical data. In "WildLife Watch," MongoDB is used to store user/watcher information, sighting reports, and other data. The database's ability to handle large volumes of data efficiently makes it ideal for the project's needs.

Fig 3.1

Fig 3.2

Fig 3.3

APIs Used

1. TensorFlow (for animal recognition)

TensorFlow is an open-source machine learning library developed by Google. It provides tools and resources for developing and training machine learning models. In "WildLife Watch," TensorFlow is used for the animal recognition feature. It processes images to identify animal species, enhancing the accuracy of sighting reports.

2. Leaflet and OpenStreetMap (for maps and location)

Leaflet: Leaflet is an open-source JavaScript library for interactive maps. In "WildLife Watch," Leaflet is used to render maps that show the locations of user sightings, national parks, wildlife sanctuaries.

CHAPTER 3

Software Requirement Specification

3.1 Product Perspective

3.1.1 System Interfaces

- The system will interface with the *MongoDB/Firebase* database to store and retrieve wildlife sighting reports, educational content, and threat alerts.
- The web interface (built with *ReactJS*) and mobile app (built with *Flutter*) will provide user-friendly interactions for reporting wildlife sightings and accessing educational resources.

3.1.2 Interfaces

- The platform will have interfaces for user registration, wildlife sighting reporting, educational content browsing, and receiving threat alerts.
- Admin interfaces will allow moderators(watchers) to manage reported sightings and content.

3.1.3 Hardware Interfaces

- The mobile app will require standard smartphone hardware (camera, microphone) for wildlife sighting reporting and user engagement.
- The Website requires a browser for visiting the page and monitoring the information.

3.1.4 Software Interfaces

- The platform will integrate with third-party services for email notifications,
authentication (e.g., OAuth), and possibly content delivery networks (CDNs) for hosting multimedia educational content.

3.1.5 Communications Interfaces

- The platform will communicate over HTTP(S) protocols for client-server interactions between the app/web frontend and backend services.
- Email notifications will be sent via SMTP or a third-party email service.

3.1.6 Memory Constraints

- The mobile app will need to manage local storage efficiently for caching data and images related to wildlife sightings.
- The web platform should optimize image and video loading to accommodate varying network speeds.

3.1.7 Operations

- Regular backups and data synchronization processes will be implemented to ensure data integrity and availability.- The platform should handle concurrent user interactions smoothly, especially during peak usage times.

3.1.8 Site Adaptation Requirements

- The platform should be adaptable to different geographical locations and ecosystems, allowing for customization based on local wildlife species and conservation needs.

3.2 Product Functions

User:

- Allow users to report wildlife sightings with time, location descriptions (if available), and media uploads like photos, name of animal, condition.
- Provide educational resources such as articles, videos, and quizzes about local wildlife and conservation practices.
- Send threat alerts and conservation updates to users based on reported sightings and ongoing conservation efforts.

Watcher:

-Allow watcher to login to the system with different ID (than users) and operate further.

-Watcher should be able to add, update, delete information of animals and also review the sightings of users.

3.3 User Characteristics

- Users may vary in technical proficiency, from casual wildlife enthusiasts to conservation experts.

- Users should be motivated by a shared interest in wildlife conservation and responsible interaction with natural habitats.

-Watcher and Officials are also part of user's scope

3.4 Constraints

- Limited or no access to GPS technology may impact the accuracy of wild life sighting reports.

- Availability of internet connectivity in remote areas may affect the real-time nature of data reporting and access.

3.5 Assumptions and Dependencies

- Assumption: Users will have access to basic smart phone devices capable of running the Flutter app.

-Initially focus on a specific forest or area for deployment to validate the platform's effectiveness and gather user feedback.

- Dependency: Successful integration with Mongo DB for data storage and retrieval.

3.6 Apportioning of Requirements

- Initially focus on a specific forest or area for deployment to validate the platform's effectiveness and gather user feedback.

- Plan for scalability and expansion to other regions based on initial deployment success.

CHAPTER 4

Specific Requirement

4.1 External Interface

User Interface (Web & Mobile App)

- The system shall provide a user-friendly interface for both web and mobile platforms to accommodate users accessing the platform via different devices..

- The user interface shall allow users to log in, report wildlife sightings, access educational resources, and receive alerts.

Database Interface

- The system shall integrate with a database system to store user information, wildlife sightings, and watcher data..

- The database interface should support CRUD (Create, Read, Update, Delete) operations for managing wildlife data.

4.2 Functions

User Login:

Users shall be able to log in securely using their credentials (ID and password).

Invalid login attempts should be handled to prevent unauthorized access.

Watcher Operations:

Watchers can add, delete, and update wildlife information in the database based on real-time observations. The system shall provide CRUD functionalities specifically tailored for watcher roles.

Wildlife Sighting Reporting :

Users can report wildlife sightings by providing details such as animal name, photo, and location (place).

Reported sightings should be stored in the database for monitoring and analysis purposes.

4.3 Performance Requirements

Real-time Data Processing:

The system should handle real-time data processing for wildlife sightings and updates from watchers.

Response times for user interactions (login, reporting sightings) should be optimized for a seamless user experience.

Scalability: The system architecture should be scalable to accommodate increasing numbers of users and wildlife data without compromising performance.

4.4 Logical Data Base Requirements

Data Entities

The logical database design should define the entities (or tables) that will store relevant data for the system:

1. User Profile Entity:

Attributes: `UserID` (Primary Key), `Name`, `Contact`, `Address`, `Password`

Description: Stores information about registered users of the platform.

2. Watcher Entity:

Attributes: `WatcherID` (Primary Key), `Name`, `Age`, `Gender`, `Contact`, `Address`, `Role`, `Password`, `PlaceID` (Foreign Key)

Description: Contains details of forest watchers who monitor wildlife and input data into the system.

3. Wildlife Sighting Entity:

Attributes: `SightingID` (Primary Key), `AnimalName`, `Photo`, `Place`, `DateTime`

Description: Stores reported wildlife sightings along with associated details.

Relationships

Define the relationships between entities to maintain data integrity and enforce business rules:

1. User-Watcher Relationship:

- Many-to-One Relationship:

- A watcher can be connected with Users.

- Foreign Key: `PlaceID` in the Watcher entity references `PlaceID` in the Place entity (if applicable).

2. User-Wildlife Sighting Relationship:

- One-to-Many Relationship:

- Each user can report multiple wildlife sightings.

- Foreign Key: `UserID` in the Wildlife Sighting entity links sightings to the corresponding watcher.

4.5 Design Constraints

4.5.1 Standards Compliance

1. Data Protection Regulations

The database design must comply with data protection laws or any other relevant local regulations regarding the protection of user data and wildlife information.

Database Security Standards

Implementation of encryption methods for sensitive data at rest and during transmission.

Access controls and user authentication mechanisms to safeguard user and watcher information.

Storage Capacity

The database design should be scalable to accommodate increasing wildlife data without compromising performance.
Constraints on the maximum size of the database due to hardware limitations should be considered.

4.6 Software System Attributes

4.6.1 Reliability

The system should maintain data integrity through error handling, validation checks, and data backups. Redundancy and failover mechanisms should be in place to mitigate data loss.

4.6.2 Availability

The platform should be highly available to users, minimizing downtime and ensuring continuous service availability. Load balancing and fault tolerance strategies should be employed to achieve high availability.

4.6.3 Security:

The platform employs robust encryption, role-based access controls, and regular security audits to protect sensitive data and maintain compliance with data protection regulations.

4.6.4 Maintainability:

Built with a modular architecture, "Wildlife Watch" supports easy updates and modifications, with well-documented code and APIs, regular code reviews, and testing.

4.6.5 Portability

The system should be compatible with different platforms (web browsers, mobile devices) to reach a broader audience. Platform-specific optimizations may be applied to ensure consistent performance across devices.

4.7 Organizing the Specific Requirements

4.7.1 System Mode

User Mode:

Users interact with the system to report wildlife sightings, access educational resources, and receive alerts.
Features accessible to users include wildlife sighting reporting and educational content browsing.

Watcher Mode:

Watchers (forest monitors) operate in this mode to add, delete, or update wildlife information in the database.
Watchers have specialized functionalities for managing wildlife data and contributing real-time observations.

4.7.2 User Class Categories:

Users:

Represents individuals registered on the platform.
Attributes include UserID, Name, Contact, Address, and Password.

Watcher:

Represents individuals registered on the platform and have some unique role like entering information of Wildlife.
Attributes include `WatcherID` (Primary Key), `Name`, `Age`, `Gender`, `Contact`, `Address`, `Role`, `Password`, `PlaceID` (Foreign Key)

4.7.3 Objects

Database Objects and Associated Services:

Key Objects:

User Object: An instance of a registered user with specific attributes (e.g., name, contact details).

Watcher Object: An instance of a forest watcher with attributes (e.g., name, age, gender) and specialized roles for wildlife monitoring.

4.7.4 Feature

Main Features:

User Login:

Allows users and watchers to authenticate securely into the system.

Wildlife Sighting Reporting:

Enables users to report wildlife sightings by providing details such as animal name, photo, and location.

Watcher Operations:

Provides watchers with functionalities to add, delete, and update wildlife information based on real-time observations.

4.7.5 Stimulus

User Interaction:

Login attempt by a user or watcher.
Submission of a wildlife sighting report.

4.7.6 Response

System Response:

Successful authentication and access granted to the appropriate mode (user or watcher).

Wildlife sighting data stored in the database and made available for analysis.

4.7.7 Functional Hierarchy

User Login Process:

Stimulus: User provides login credentials.
Response: System verifies credentials and grants access based on user role (user or watcher). Wildlife Sighting Reporting:
Stimulus: User submits wildlife sighting report.
Response: System stores the report in the database and may trigger alerts or notifications based on the reported sighting.

CHAPTER 5

Change Management Process

The Change Management Process within the Software Requirements Specification (SRS) for the Wildlife Watch System is crucial for adapting to evolving needs and ensuring system success. It involves identifying, evaluating, approving, and implementing proposed changes to system requirements efficiently. Stakeholders, including users, domain experts, and technical teams, participate in proposing and evaluating changes based on impact analysis, feasibility, and alignment with project goals. A team oversees the approval process, ensuring transparent decision-making. Implementing approved changes involves coordination among development, testing, and deployment teams, with thorough documentation and communication to stakeholders. The process includes monitoring the impact of changes on system performance and user satisfaction. By embracing change management, the system gains adaptability, risk mitigation, stakeholder engagement, and opportunities for continuous improvement, aligning with wildlife conservation objectives and community empowerment efforts. This systematic approach fosters resilience and responsiveness in addressing evolving requirements and technological advancements.

CHAPTER 6

DESIGN PHASE

6.1 ARCHITECTURAL DIAGRAM

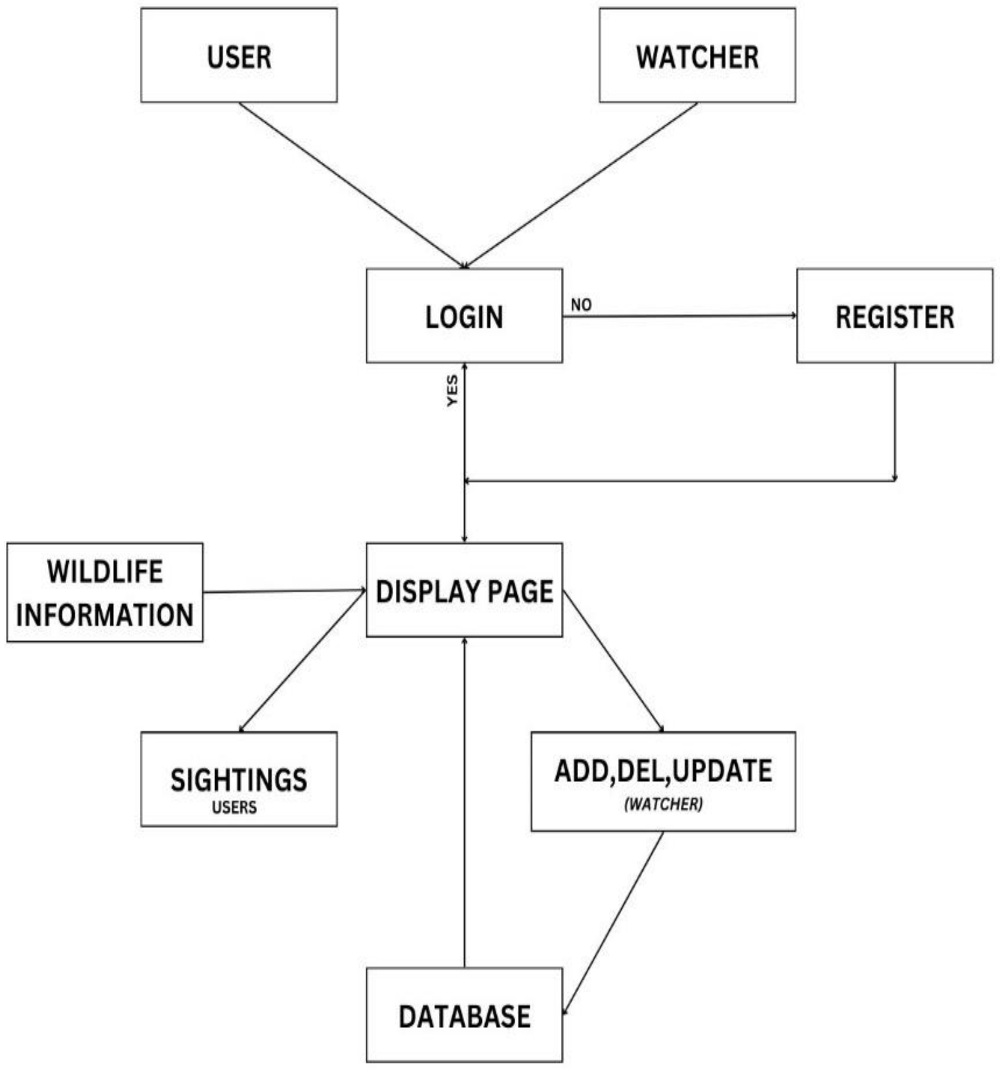


Fig 6.1

Architectural design of our project (Fig 6.1) depicts various modules like login, register, report and view sightings and data handling module through which a watcher can add, delete and modify the information related to wildlife.

6.2 DATA FLOW DIAGRAM

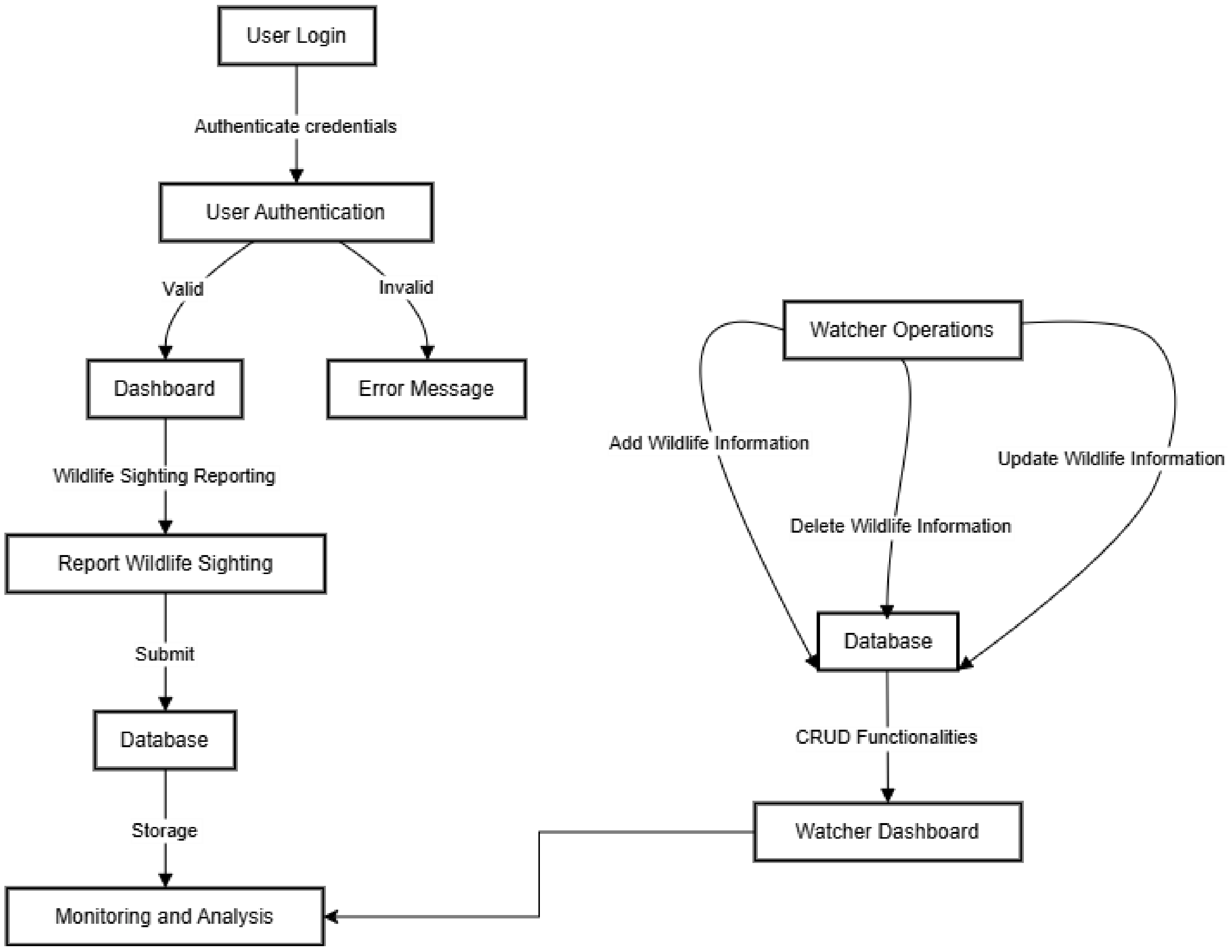


Fig 6.2

Data Flow Diagrams (Fig 6.2) depicts how data is transferred through various modules and get stored. Information like login / signup details, sightings details, watcher and related details are stored and managed efficiently.

6.3 CLASS DESIGN

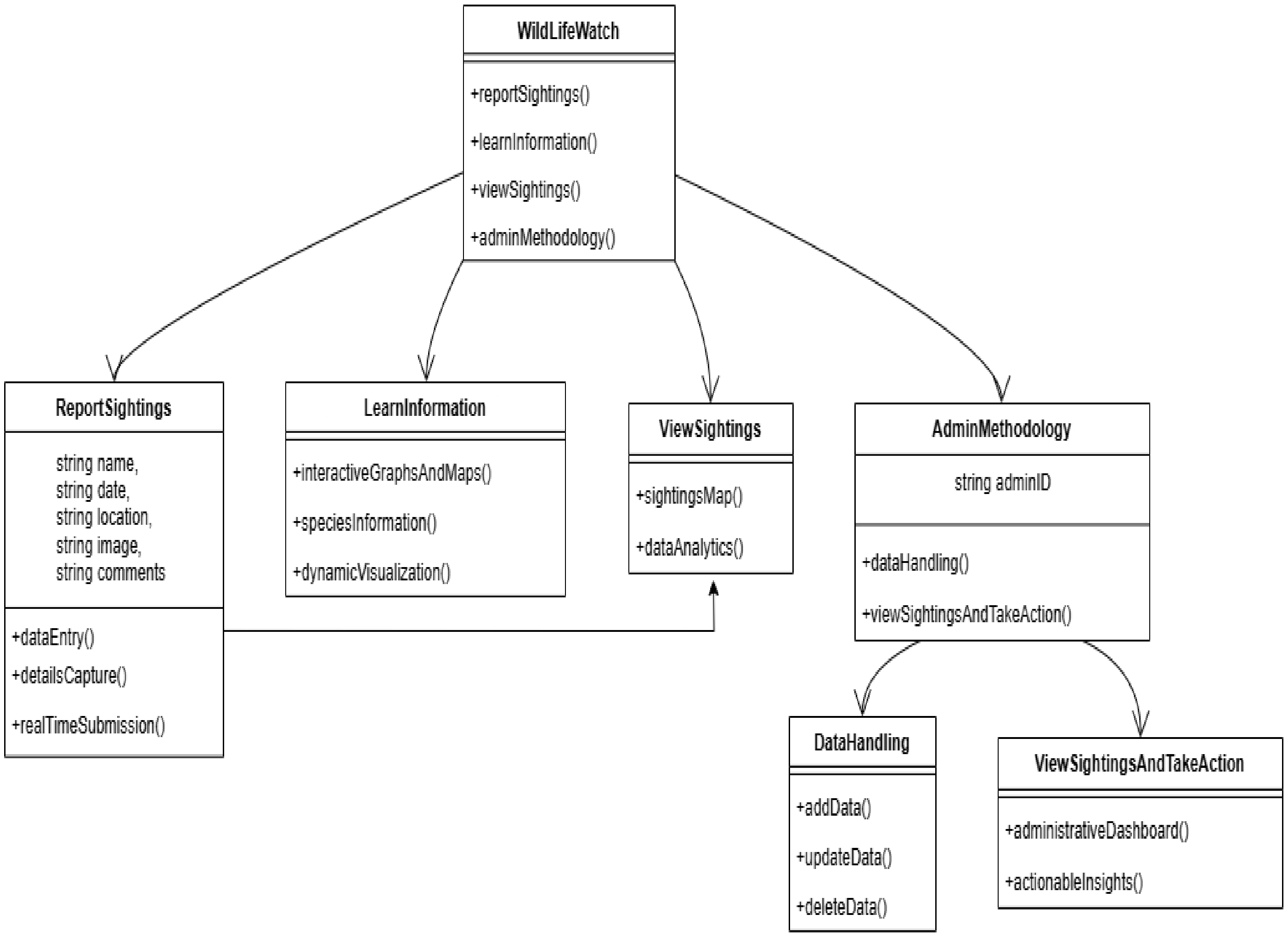


Fig 6.3

Class diagram (Fig 6.3) provides an in-depth overview of the main classes and their methods related to the functionalities of our project "Wild Life Watch".

6.4 USE CASE DESIGN

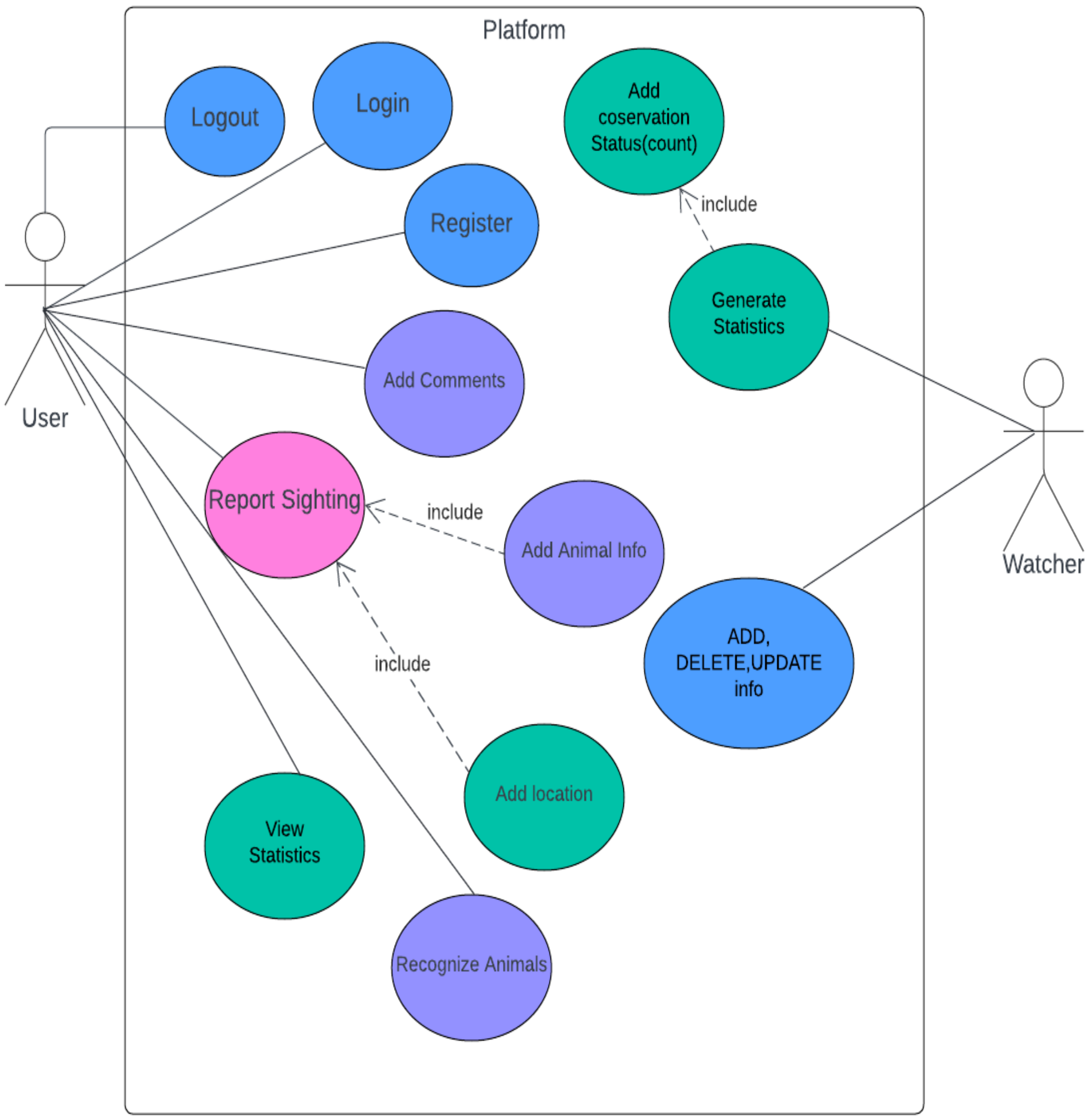


Fig 6.4

The use case diagram (Fig 6.4) illustrates the interactions between actors (users and administrators) and the system, having use cases like report sightings, learn information about wildlife, view sightings, admin methodology include data handling, view sightings and take actions.

6.5 SEQUENCE DESIGN

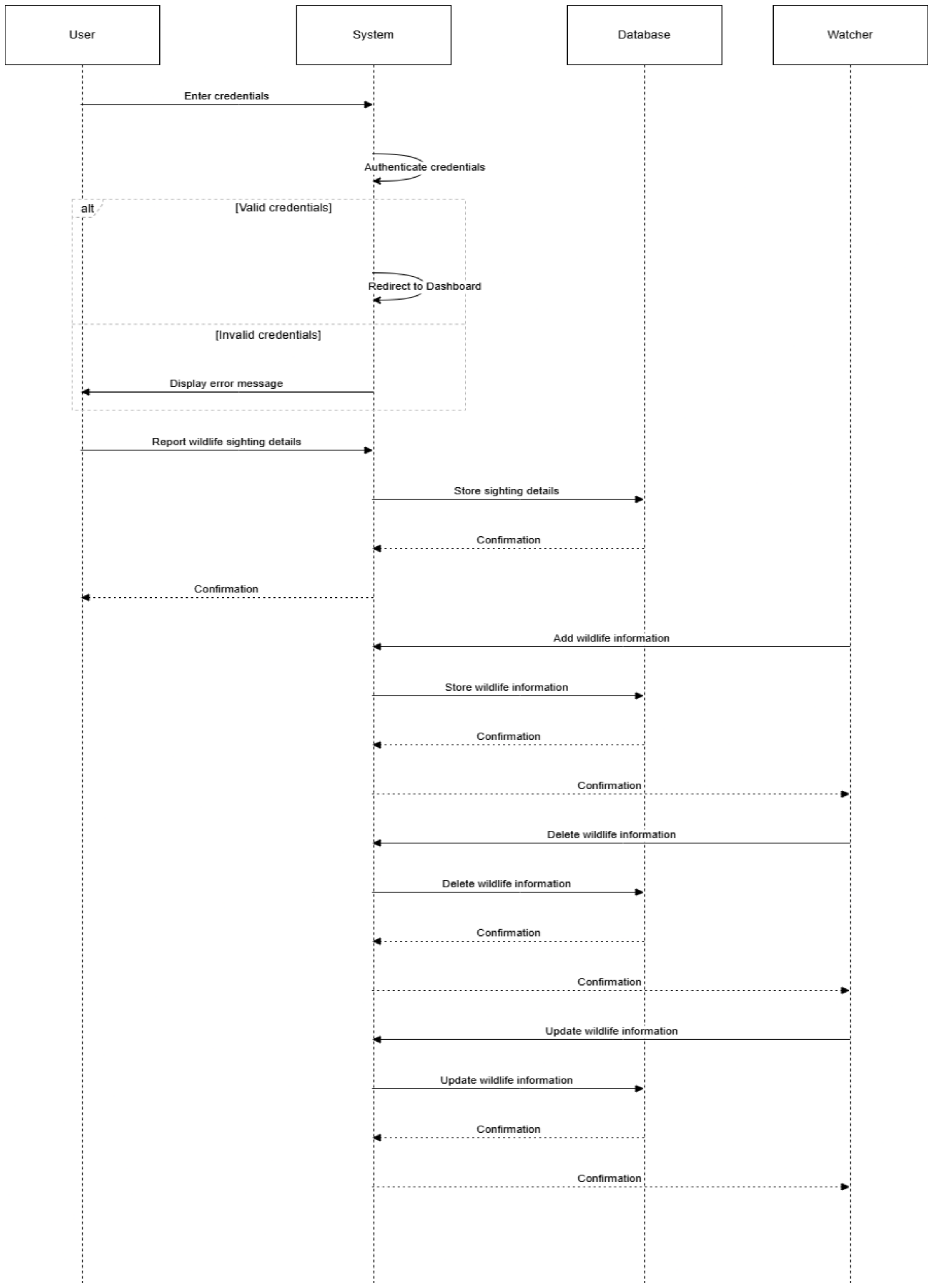


Fig 6.5

CHAPTER 7

IMPLEMENTATION PHASE

7.1Experimanetal set up

In developing "Wildlife Watch," a comprehensive web application aimed at revolutionizing wildlife monitoring and conservation efforts, a combination of frontend and backend technologies has been carefully selected to ensure optimal performance and user experience. On the frontend, HTML forms the backbone of the application, providing the basic structure and layout of web pages. CSS is then utilized to stylize and design HTML elements, enhancing visual appeal and user engagement. JavaScript, a versatile programming language, adds interactivity to the web pages, facilitating functionalities such as form validation, asynchronous data fetching, DOM manipulation, and the integration of interactive maps. Moreover, Flutter, an open-source UI software development kit by Google, has been employed for frontend purposes, enabling the creation of cross-platform applications from a single codebase. Flutter's widget-based approach allows for the seamless composition of user interfaces, ensuring consistency and efficiency across various platforms.

In parallel, on the backend, Node.js serves as the foundation for server-side operations, enabling the execution of JavaScript code outside of web browsers. Leveraging its scalability and efficiency, Node.js handles request processing, interacts with the database, and manages server-side functionalities within the "Wildlife Watch" application. Complementing Node.js, Express.js, a web application framework, provides a streamlined development experience by offering robust features for routing, middleware integration, and HTTP request handling. With Express.js, developers can define server routes, manage middleware for request processing, and orchestrate server-side logic seamlessly. Meanwhile, MongoDB, a NoSQL database, serves as the backends' data storage solution. Renowned for its flexibility and scalability, MongoDB efficiently stores user information, wildlife sightings, and other relevant data in JSON-like documents, catering to the project's data management needs effectively.
By integrating these frontend and backend technologies, "Wildlife Watch" delivers a user-centric and scalable platform for wildlife monitoring and conservation. The frontend technologies ensure an intuitive and visually appealing user interface, while the backend technologies empower the application with robust server-side capabilities and efficient data management functionalities.

7.2 Pseudo code

```
 → Initialize System
    - Load necessary libraries and dependencies
    - Set up database connection
```

Code:

```
// MongoDB connection
mongoose.connect('mongodb://localhost:27017/Simple_database',
{use NewUrlParser :true, useUnifiedTopology: true });
const db = mongoose.connection;
db.on('error', console.error.bind(console, "Error in connecting to Database"));
db.once('open', () => console.log("Connected to Database"));
```

```
→ User Authentication
    - Prompt user to login or signup
    - Validate user credentials
    - Provide options for forgotten password recovery
```

```
// User login endpoint
app.post("/user_login", async (req, res) => {
    try {
        const { email, password } = req.body;
        const user = await db.collection('login_details').findOne({ email, password });

        if (user) {
            res.redirect('welcome.html');
        } else {
            res.send("Invalid Login Credentials");
        }
    } catch (error) {
        console.error('Error logging in:', error);
        res.status(500).send('Failed to log in. Please try again later.');
    }
});

// Admin login endpoint
app.post("/admin_login", async (req, res) => {
  try {
    const { adminID, admin_name, password } = req.body;
    const admin = await db.collection('admin_details').findOne({ adminID });

    if (admin && admin.username === admin_name && admin.password === password) {
      res.redirect('welcome_admin.html');
    } else {
      res.send("Invalid Login Credentials");
    }
  } catch (error) {
    console.error('Error logging in as admin:', error);
    res.status(500).send('Failed to log in. Please try again later.');
  }
});
```

→User Dashboard

- Display user-specific information and options
- Allow user to report wildlife sightings
- Provide access to educational resources
- Display real-time wildlife sightings in user's area
- Option to view and edit user profile

```
//report sighting
<section class="report-section">
      <div class="container">
        <div style="text-align: center;">
          <h1>Animal Sighting Report</h1>
        </div>
        <form action="report-sighting" method="POST" enctype="multipart/form-data">
```

```
            <label for="userName">Name:</label>
            <input type="text" id="userName" name="userName" placeholder="Enter your name"
required>

            <label for="userEmail">Email:</label>
            <input type="email" id="userEmail" name="userEmail" placeholder="Enter your emailId"
required>

            <label for="sightingDate">Date of Sighting:</label>
            <input type="date" id="sightingDate" name="sightingDate" required>

            <label for="uploadPic">Upload Image:</label>
            <input type="file" id="uploadPic" name="uploadPic" accept="image/*" required>
            <img id="previewImage" src="#" alt="Image Preview"
              style="display:none; width: 200px; height: auto; margin-top: 10px;">

            <label for="location">Location of Sighting:</label>
            <div id="map" style="height: 400px;"></div>
            <div id="coordinates">
              <label for="latitude">Latitude:</label>
              <input type="text" id="latitude" name="latitude" readonly>
              <label for="longitude">Longitude:</label>
              <input type="text" id="longitude" name="longitude" readonly>
              <label for="placeName">Place Name:</label>
              <input type="text" id="placeName" name="placeName">
            </div>
            <label for="comments">Additional comments:</label>
            <textarea name="comments" id="comments" placeholder="anything to say..."></textarea>

            <button type="submit">Report Sighting</button>
          </form>
        </div>
        </section>
```

→Report Wildlife Sighting

- Collect sighting details from user (species, location, time, etc.)
- Validate and store sighting data in the database
- Generate confirmation message to user

```
// Report sighting endpoint
app.post("/sightings/report-sighting", upload.single('uploadPic'), async (req, res) => {
    try {
        const { userName, userEmail, sightingDate, latitude, longitude, placeName, comments } =
req.body;
        const imagePath = `/uploads/${req.file.filename}`;
        const sightingData = {
            name: userName,
            email: userEmail,
            date: sightingDate,
            imagePath,
            latitude,
            longitude,
            location: placeName,
            comments
        };

        await db.collection('sighting_details').insertOne(sightingData);
        console.log(sightingData);
        console.log("Sighting report stored in database!");

        // Send SMS notification to users matching the sighting location
        // const sightings = await db.collection('sighting_details').find({ name: userName }).toArray();
        const sightings = await db.collection('sighting_details').findOne({ location: placeName });
        const users = await db.collection('login_details').find({ location: placeName }).toArray();
        users.forEach(async user => {
            await twilioClient.messages.create({
                body: `Hello ${user.name}, there has been an animal sighting at ${sightings.location} on
${sightings.date}.`,
                from: '+13204296656', // Replace with your Twilio phone number
                to: `+91${user.mobile}`
            });
        });

        res.status(200).send("Sighting reported successfully!");
    } catch (error) {
        console.error('Error reporting sighting:', error);
        res.status(500).send('Failed to report sighting. Please try again later.');
    }
});
```

→View Wildlife Data

- Retrieve and display wildlife sightings from the database
- Filter sightings based on user preferences (species, location, time, etc.)
- Visualize data using interactive maps and charts

```
//display sightings
<script>
            // Fetch sightings data
            fetch('http://localhost:3000/sightings')
                .then(response => response.json())
                .then(data => {
                    const sightingsDiv = document.getElementById('sightings');
                    data.forEach(sighting => {
                        const sightingElement = document.createElement('div');
                        sightingElement.classList.add('col-md-4'); // Add Bootstrap column class
                        sightingElement.innerHTML = `
                        <div class="card mb-4" style="width: 18rem;">
                    <img src="${sighting.imagePath}" class="card-img-top w-100" alt="Sighting Image"
>
                    <div class="card-body">
                        <p class="card-text">
                          <b> ${sighting.name} </b><br>
                           <i>Email:</i> ${sighting.email}<br>
                           <i>Date:</i> ${sighting.date}<br>
                           <i>Place:</i>                           ${sighting.location}(${sighting.latitude},
${sighting.longitude})<br>
                           <i>Comments:</i> ${sighting.comments}
                        </p>
                    </div>
                </div> `;
                sightingsDiv.appendChild(sightingElement);
            });
        })
        .catch(error => console.error('Error fetching sightings:', error));
    </script>
```

→Educational Resources

- Provide access to articles, videos, and interactive content about wildlife and conservation
- Categorize resources for easy navigation
- Option to search for specific topics or keywords

→Notification System

- Send real-time notifications to users about wildlife sightings in their area
- Allow users to customize notification preferences (frequency, species, location, etc.)

```
// Send SMS notification to users matching the sighting location
    const sightings = await db.collection('sighting_details').findOne({ location: placeName });
    const users = await db.collection('login_details').find({ location: placeName }).toArray();
    users.forEach(async user => {
        await twilioClient.messages.create({
            body: `Hello ${user.name}, there has been an animal sighting at ${sightings.location} on ${sightings.date}.`,
            from: '+13204296656',
            to: `+91${user.mobile}`
          });
```

→ Data Analysis and Insights

- Analyze wildlife sighting data to identify trends and patterns
- Generate reports and insights for researchers and conservationists
- Provide recommendations for conservation actions based on data analysis

→ Security and Privacy

- Implement encryption and secure protocols for user authentication and data transmission
- Enforce access controls and permissions to protect sensitive information
- Regularly update and audit security measures to ensure compliance with data protection regulations

Error Handling and Exception Management

- Implement error handling mechanisms to gracefully handle unexpected situations
- Provide informative error messages to users and log errors for troubleshooting

→Logging and Monitoring

- Log system activities and user interactions for audit and analysis purposes
- Monitor system performance and resource usage to optimize efficiency and scalability

→Close System

- Clean up resources and close database connection
- Log out users and terminate system operation

CHAPTER 8
TESTING PHASE

8.1 Types of tests carried out

The testing phase for "Wildlife Watch" is crucial to ensure the reliability, performance, and security of the platform. During this phase, a series of rigorous tests are conducted to identify and rectify any issues before the system is deployed to end-users.

- Unit Testing: Each individual component and module of the system is tested independently to verify that they function correctly in isolation. Automated unit tests are employed to ensure that the core functionalities perform as expected and that any changes or updates do not introduce new bugs.

- Integration Testing: Once the individual components are verified, integration testing is performed to ensure that different modules work together seamlessly. This involves testing interactions between the backend services, APIs, and the user interface to ensure data flows correctly across the system.

- System Testing: The complete system is tested as a whole to ensure it meets the specified requirements. This includes end-to-end testing of all features such as user registration, login, wildlife sighting reporting, notifications, and data visualization. Both functional and non-functional requirements are validated during this phase.

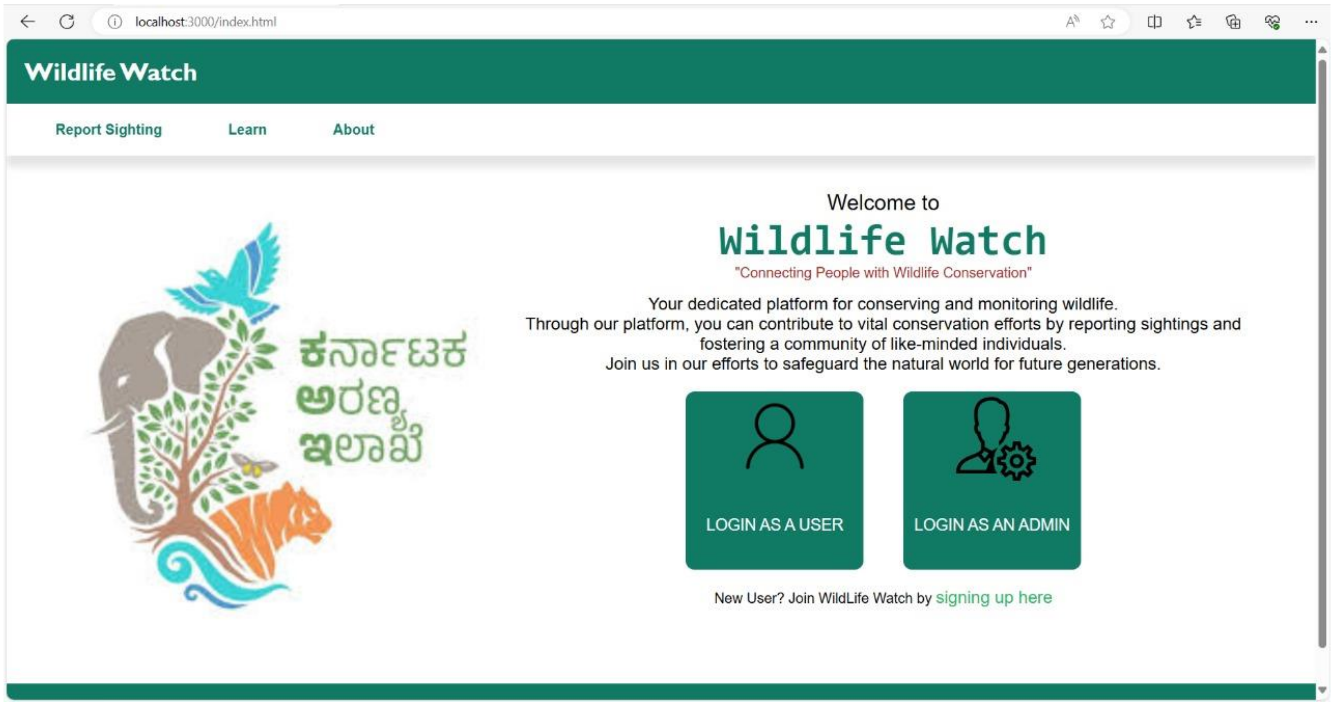
- Performance Testing: Performance tests are conducted to evaluate the system's responsiveness and stability under various conditions. This includes load testing to determine how the system handles high volumes of user activity and stress testing to identify its breaking point. Scalability is also tested to ensure the platform can grow to accommodate more data and users over time.

- Security Testing: Given the sensitive nature of wildlife data and user information, security testing is paramount. Penetration tests and vulnerability assessments are conducted to identify potential security flaws and ensure robust protection against unauthorized access. Encryption methods and access controls are verified to uphold data integrity and confidentiality.

- Regression Testing: After any bug fixes or new features are implemented, regression testing is conducted to ensure that the existing functionalities continue to work correctly. Automated regression tests help in quickly validating that updates do not negatively impact the system.

CHAPTER 9
RESULTS & DISCUSSIONS

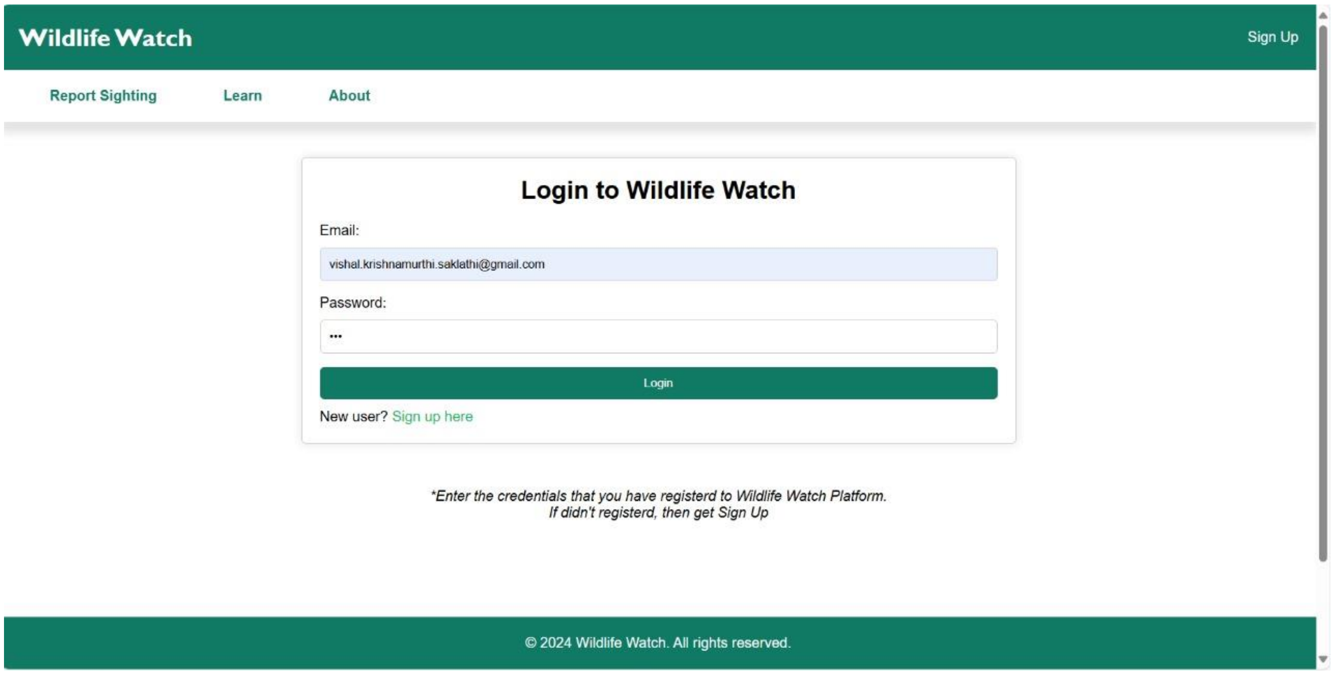
9.1 Login Page



Welcome to
Wildlife Watch
"Connecting People with Wildlife Conservation"

Your dedicated platform for conserving and monitoring wildlife.
Through our platform, you can contribute to vital conservation efforts by reporting sightings and fostering a community of like-minded individuals.
Join us in our efforts to safeguard the natural world for future generations.

New User? Join WildLife Watch by signing up here



Login to Wildlife Watch

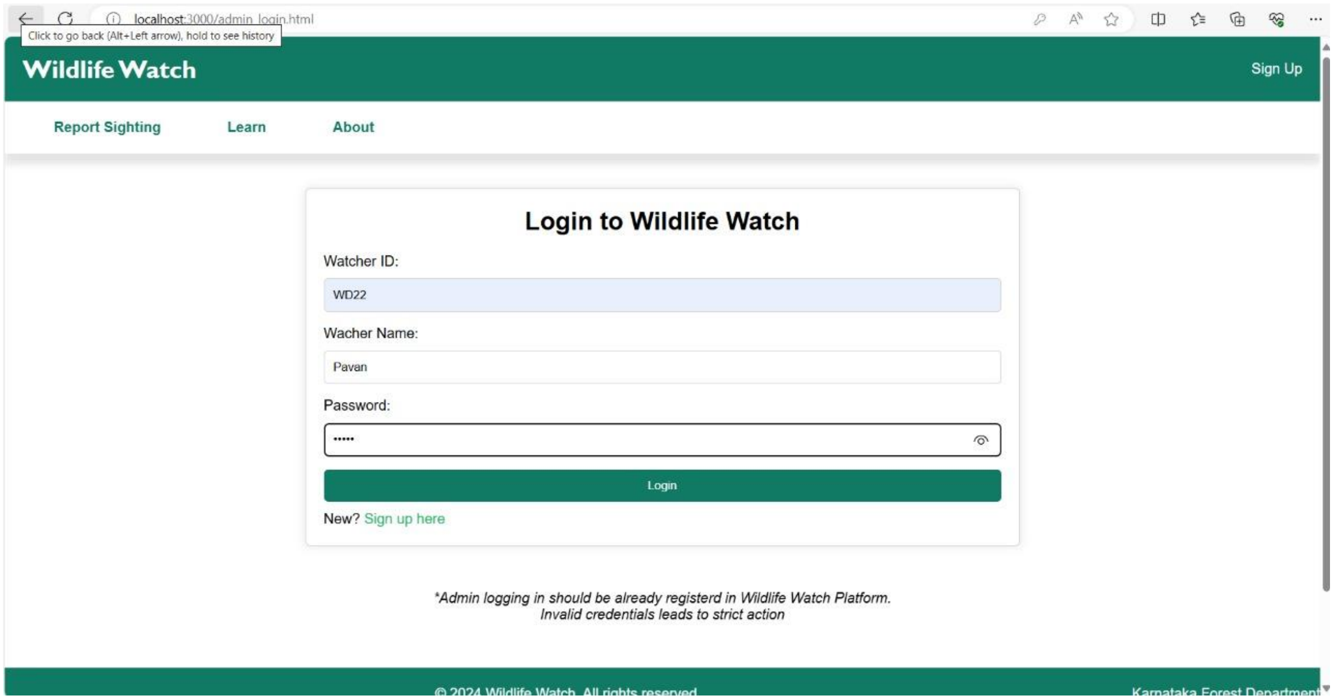
Email: vishal.krishnamurthi.saklathi@gmail.com

Password: •••

New user? Sign up here

*Enter the credentials that you have registerd to Wildlife Watch Platform.
If didn't registerd, then get Sign Up

© 2024 Wildlife Watch. All rights reserved.



localhost:3000/admin_login.html

Click to go back (Alt+Left arrow), hold to see history

Wildlife Watch

Sign Up

Report Sighting Learn About

Login to Wildlife Watch

Watcher ID: WD22

Wacher Name: Pavan

Password: •••••

Login

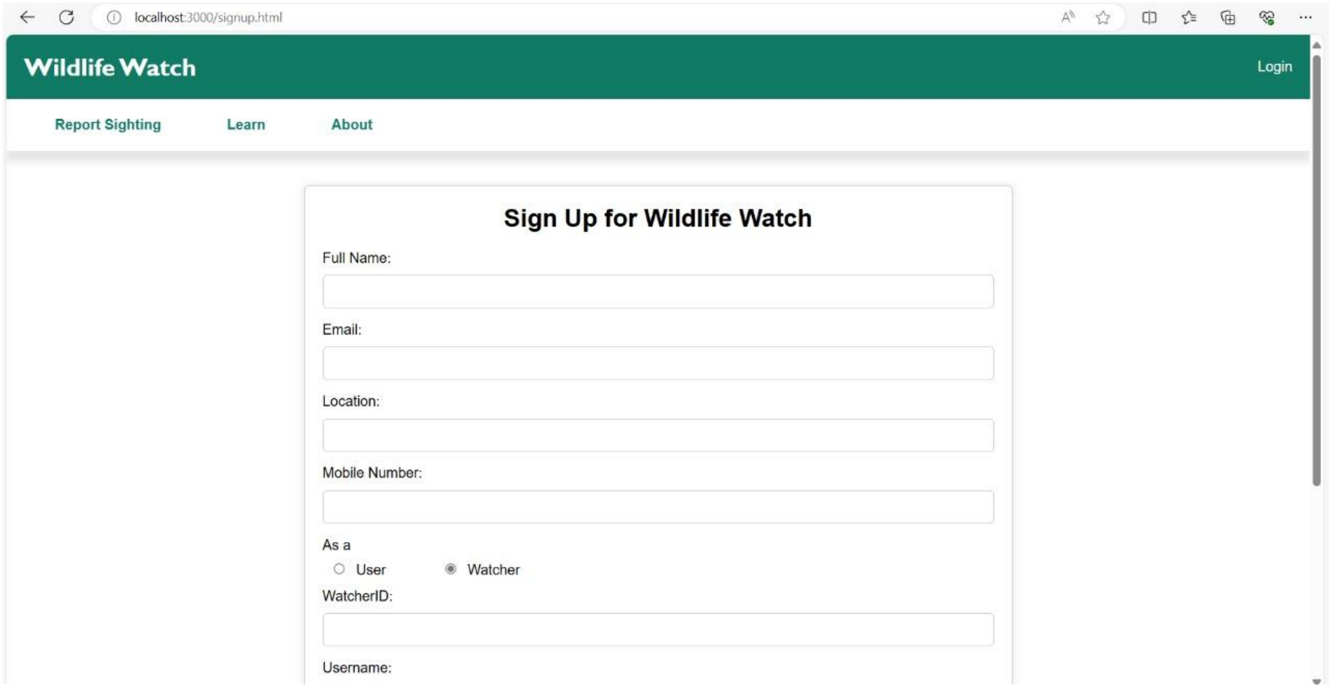
New? Sign up here

*Admin logging in should be already registerd in Wildlife Watch Platform.
Invalid credentials leads to strict action

© 2024 Wildlife Watch. All rights reserved. Karnataka Forest Department

Login System has two modules Login as user and Login as a watcher, and those who are logging in to the WildLife Watch must have an account or else should create new account.

9.2 Sign Up Page



localhost:3000/signup.html

Wildlife Watch

Login

Report Sighting Learn About

Sign Up for Wildlife Watch

Full Name:

Email:

Location:

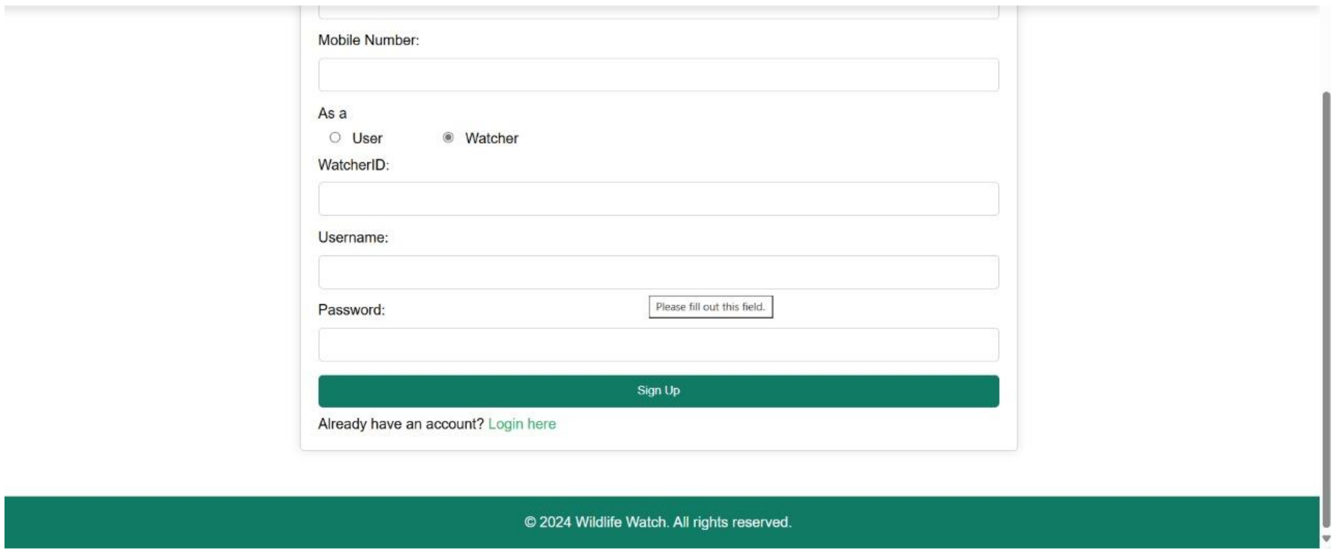
Mobile Number:

As a

User [✓ Watcher]

WatcherID:

Username:


Mobile Number:

As a

User [✓ Watcher]

WatcherID:

Username:

Password:

Please fill out this field.

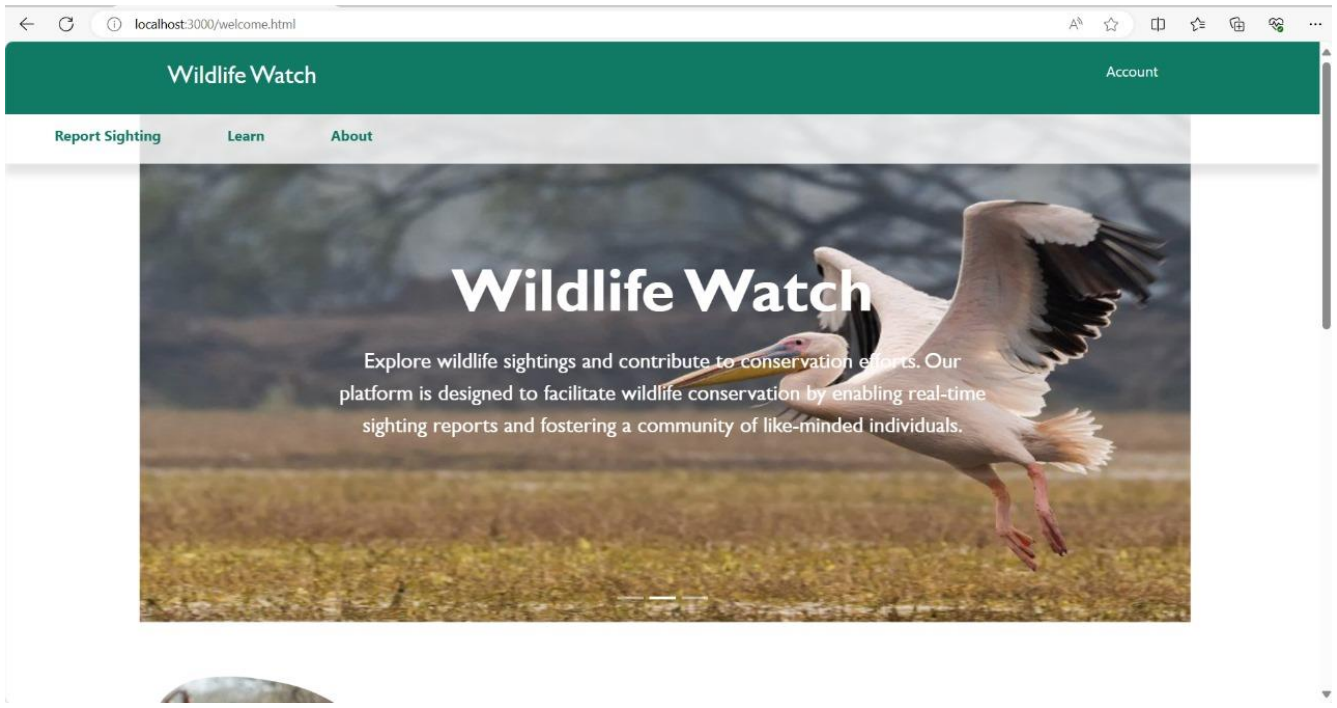
Sign Up

Already have an account? Login here

© 2024 Wildlife Watch. All rights reserved.

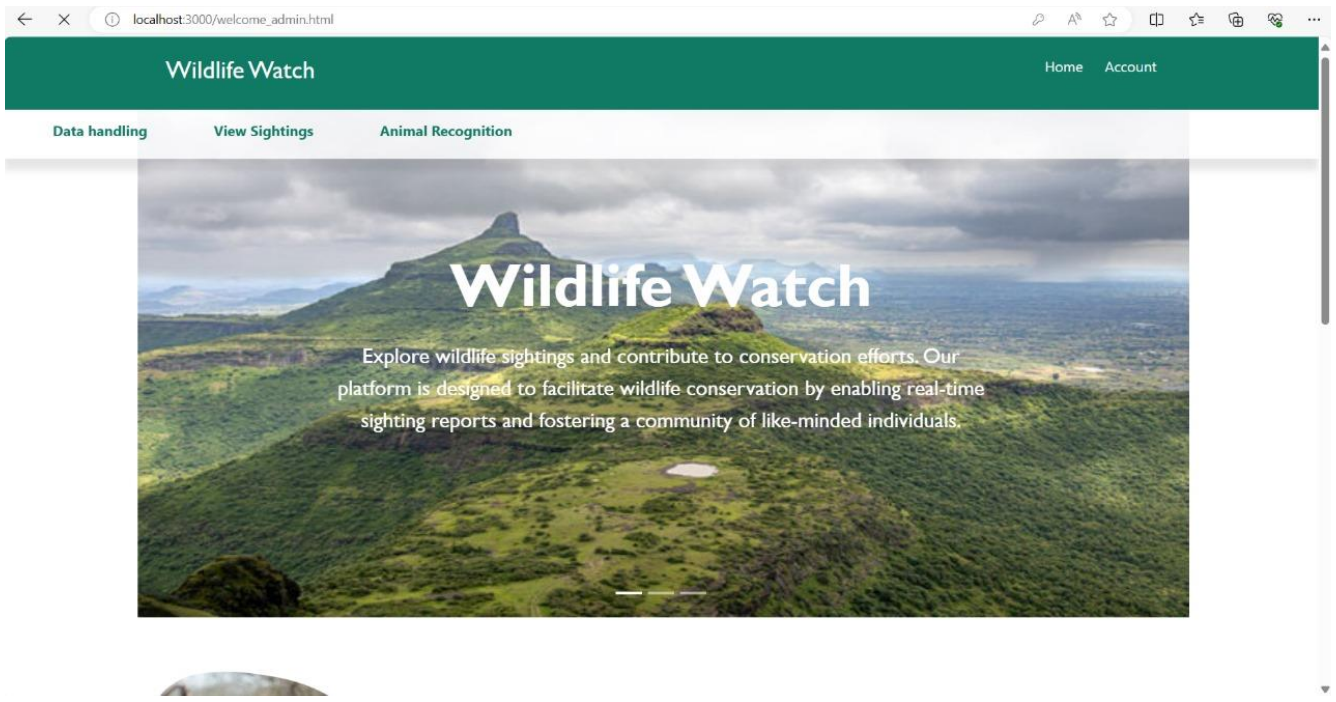
By filling up necessary information, one can create account in WildLife Watch. And access various functions of website.

9.3Home Page (User)



Explore wildlife sightings and contribute to conservation efforts. Our platform is designed to facilitate wildlife conservation by enabling real-time sighting reports and fostering a community of like-minded individuals.

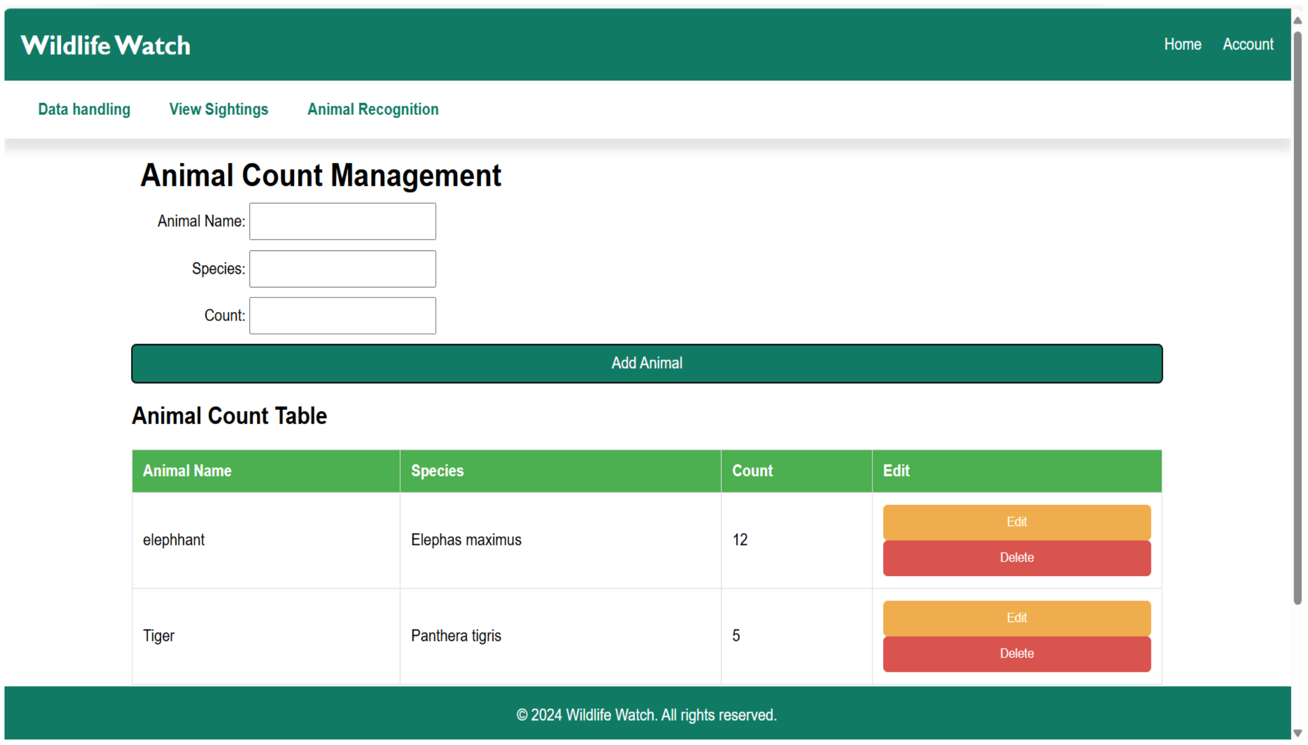
9.4 Home Page (Watcher)



Explore wildlife sightings and contribute to conservation efforts. Our platform is designed to facilitate wildlife conservation by enabling real-time sighting reports and fostering a community of like-minded individuals.

As one can see homepage has good user interface.

9.5 Animal Data Management



Animal Count Management

Animal Name:	
Species:	
Count:	

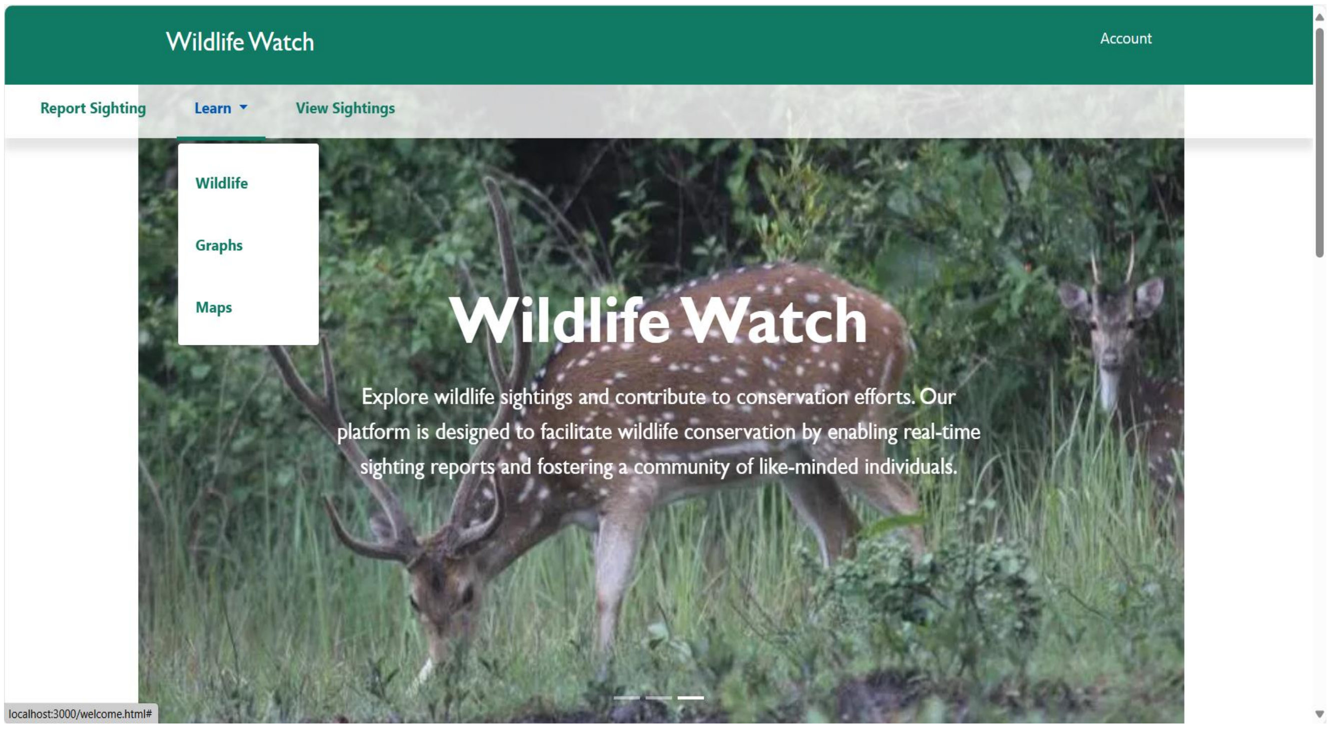
Animal Count Table

Animal Name	Species	Count	Edit
elephhant	Elephas maximus	12	Edit Delete
Tiger	Panthera tigris	5	Edit Delete

© 2024 Wildlife Watch. All rights reserved.

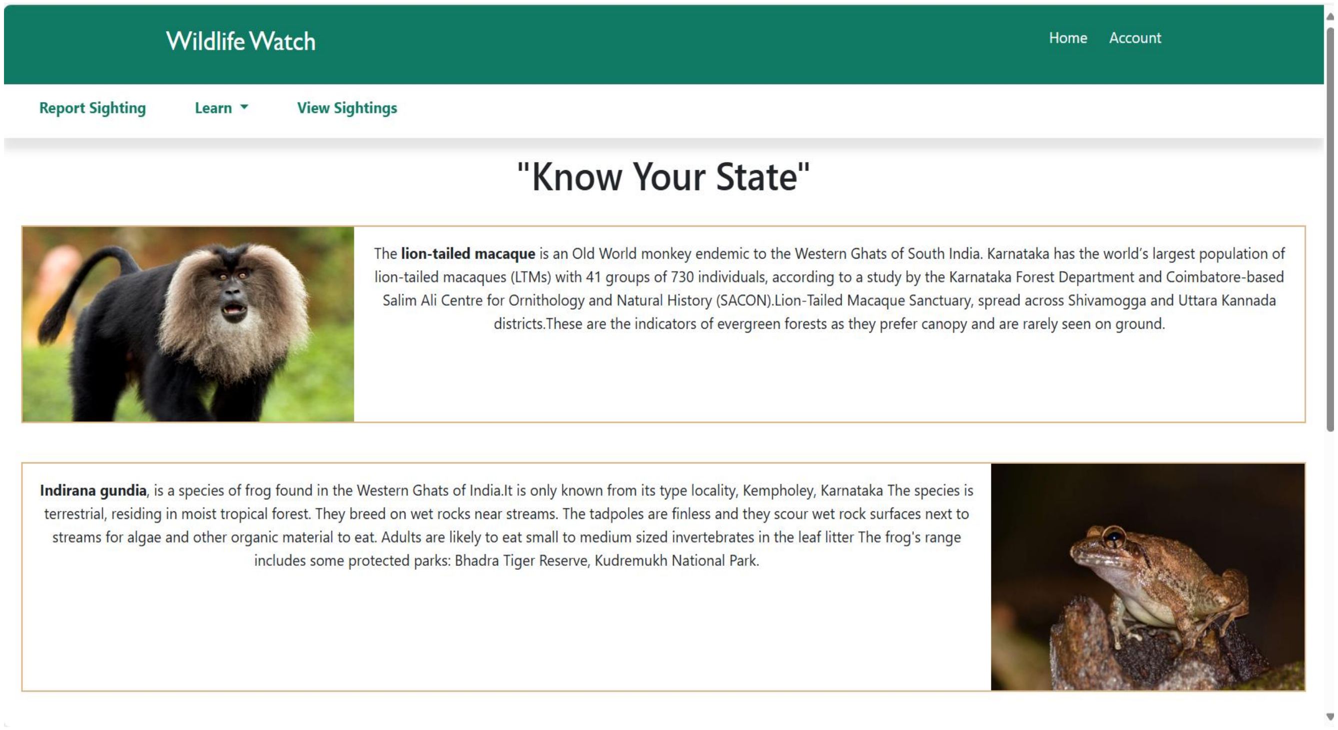
This module is managed by watcher to add information about wildlife.

9.6 Learn Section



Wildlife Watch

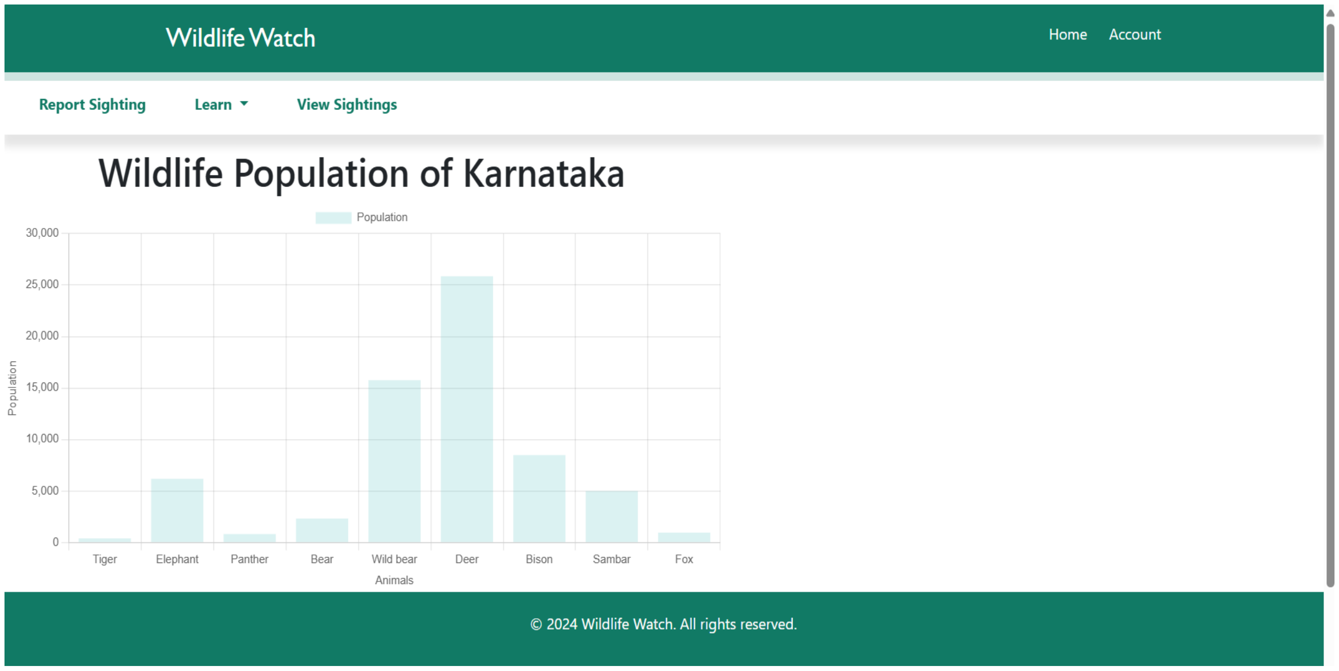
Explore wildlife sightings and contribute to conservation efforts. Our platform is designed to facilitate wildlife conservation by enabling real-time sighting reports and fostering a community of like-minded individuals.



"Know Your State"

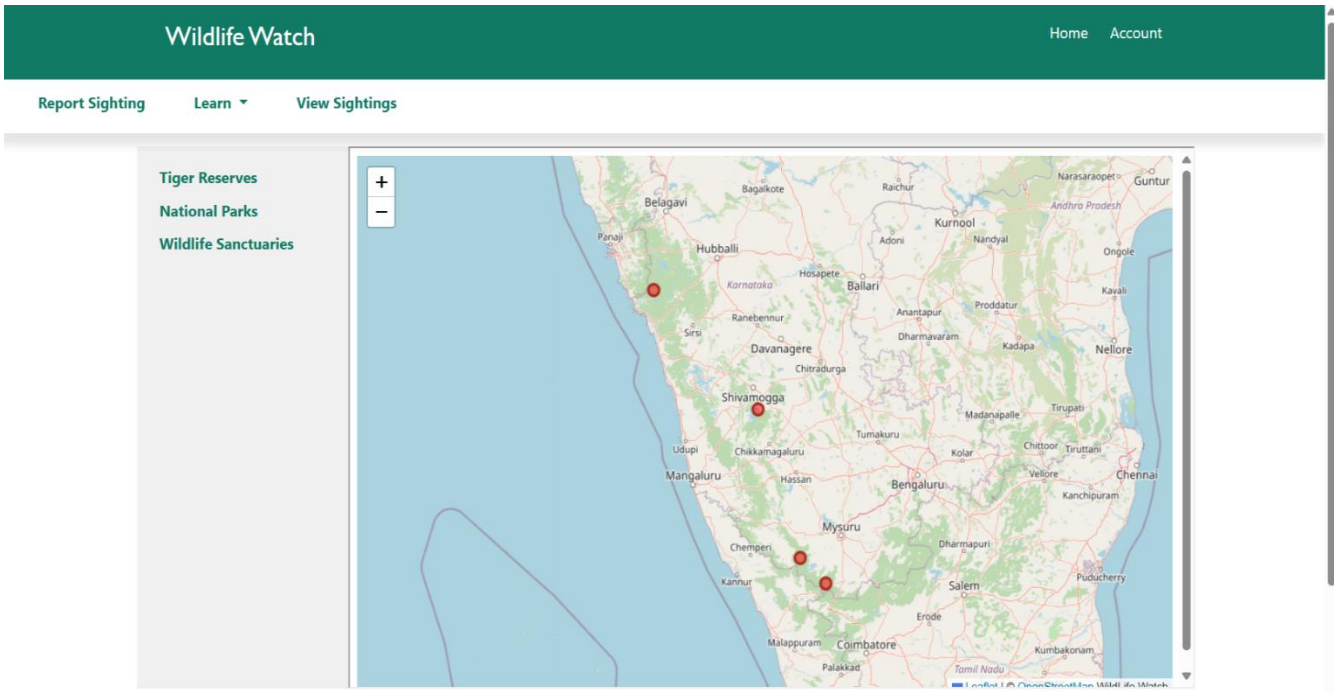
The **lion-tailed macaque** is an Old World monkey endemic to the Western Ghats of South India. Karnataka has the world's largest population of lion-tailed macaques (LTMs) with 41 groups of 730 individuals, according to a study by the Karnataka Forest Department and Coimbatore-based Salim Ali Centre for Ornithology and Natural History (SACON).Lion-Tailed Macaque Sanctuary, spread across Shivamogga and Uttara Kannada districts.These are the indicators of evergreen forests as they prefer canopy and are rarely seen on ground.

Indirana gundia, is a species of frog found in the Western Ghats of India.It is only known from its type locality, Kempholey, Karnataka The species is terrestrial, residing in moist tropical forest. They breed on wet rocks near streams. The tadpoles are finless and they scour wet rock surfaces next to streams for algae and other organic material to eat. Adults are likely to eat small to medium sized invertebrates in the leaf litter The frog's range includes some protected parks: Bhadra Tiger Reserve, Kudremukh National Park.



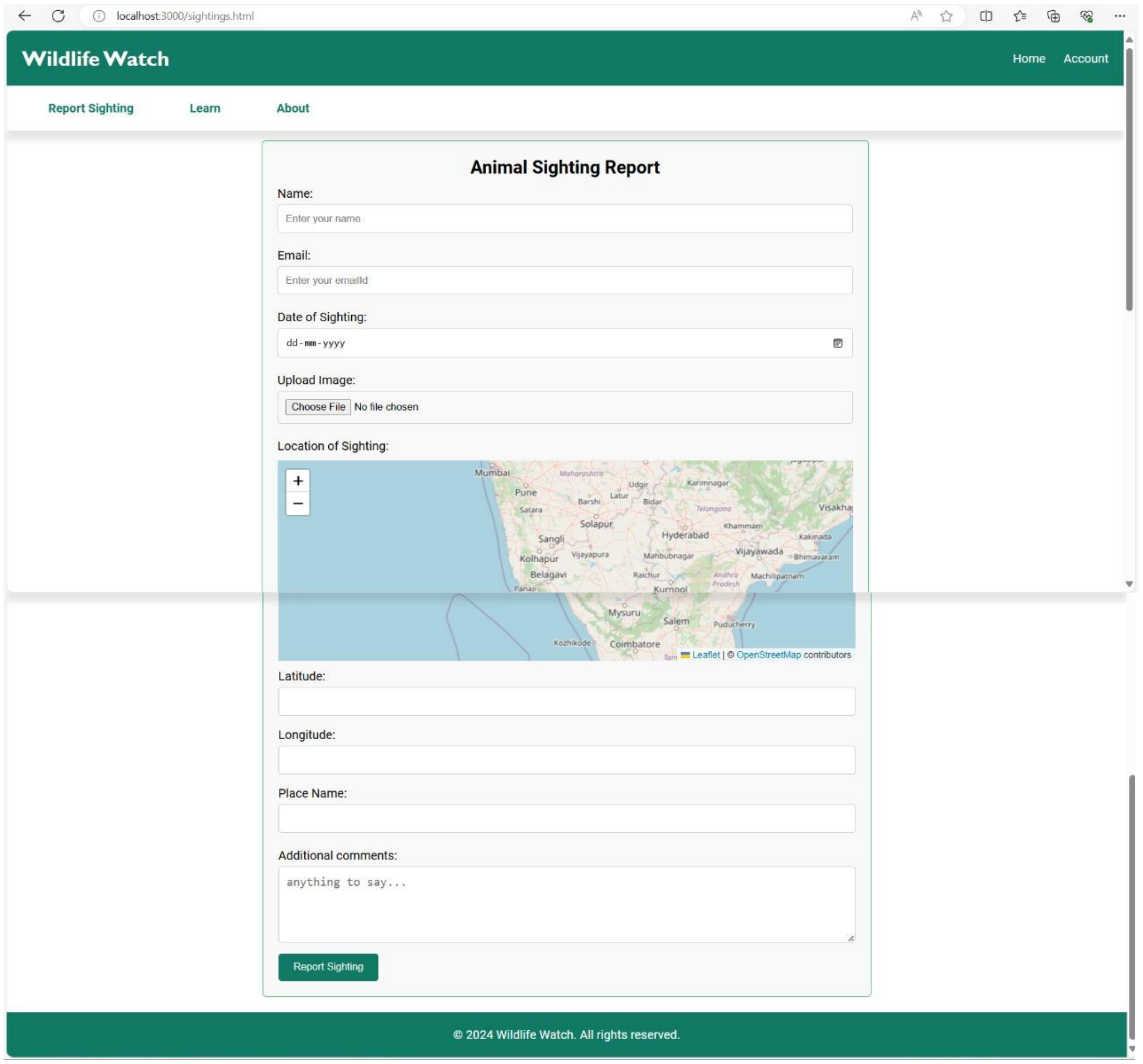
Wildlife Population of Karnataka

© 2024 Wildlife Watch. All rights reserved.



Learning Section provides information about wildlife in the Karnataka State. It is able to engage users via interactive maps and graphs.

9.7 Report Sighting


localhost:3000/sightings.html

Wildlife Watch

Home Account

Report Sighting Learn About

Animal Sighting Report

Name: Enter your name

Email: Enter your emailId

Date of Sighting: dd-mm-yyyy

Upload Image: Choose File No file chosen

Location of Sighting:

Latitude:

Longitude:

Place Name:

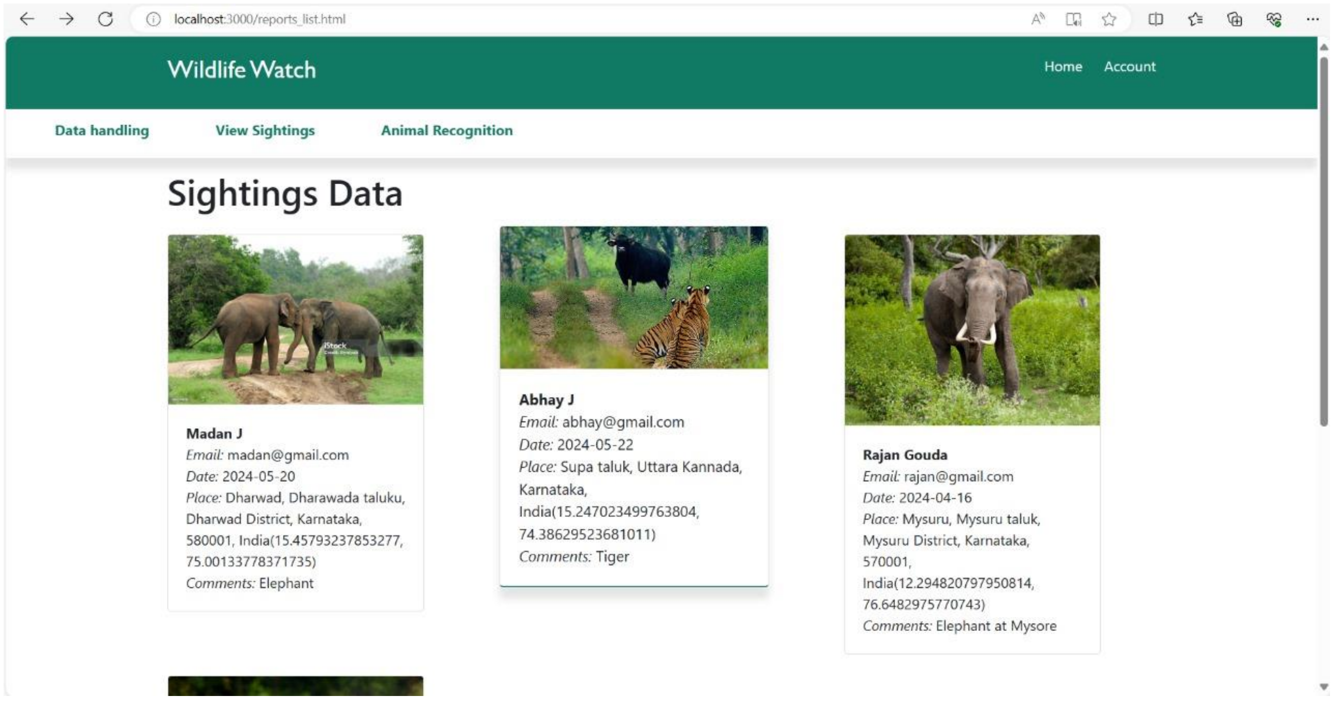
Additional comments: anything to say...

Report Sighting

© 2024 Wildlife Watch. All rights reserved.

Report Sighting is the main module if the WildLife Watch. Here one can report a sighting of animals in some crowded places, or report poaching of animals, injured animals. One has to enter his name, email ID, date of sightings, upload a photo of animal, and select location and add further comments if necessary to report a sighting. These sightings are later viewed by watcher and actions will be taken.

9.8 View Sightings



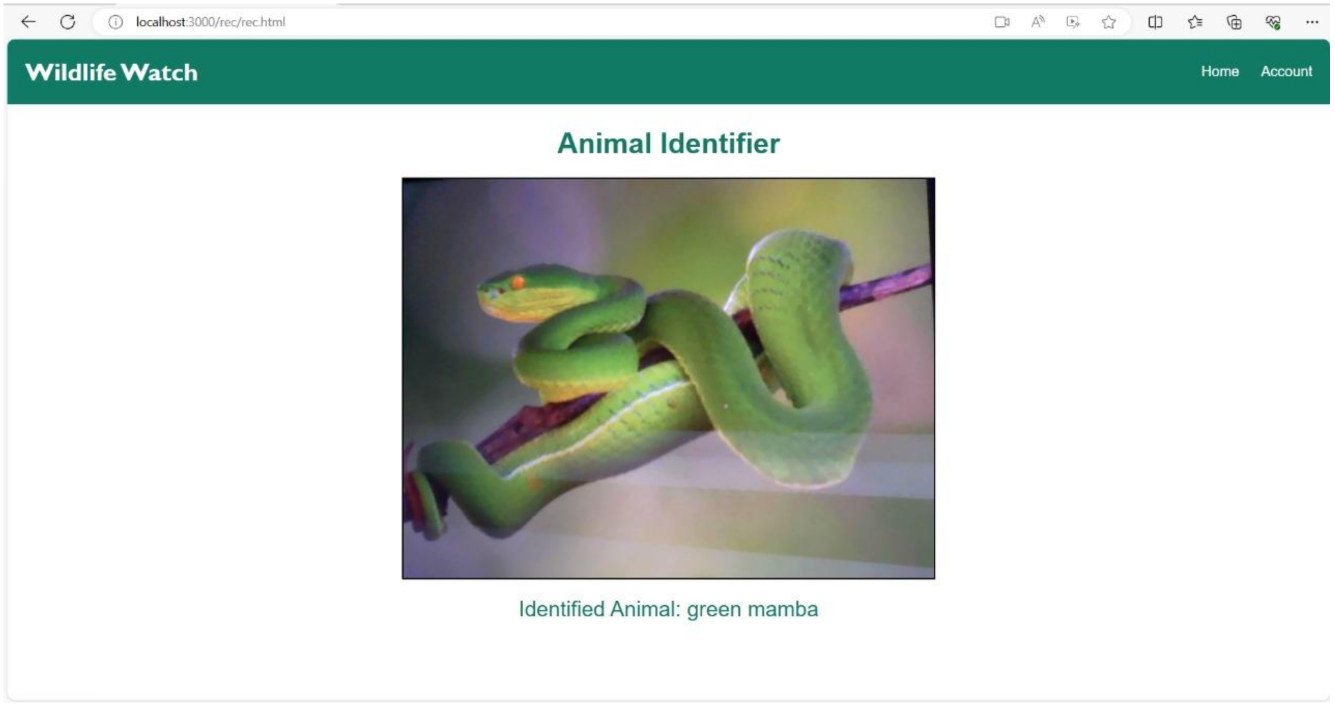
Sightings Data

Madan J
Email: madan@gmail.com
Date: 2024-05-20
Place: Dharwad, Dharawada taluku, Dharwad District, Karnataka, 580001, India(15.45793237853277, 75.00133778371735)
Comments: Elephant

Abhay J
Email: abhay@gmail.com
Date: 2024-05-22
Place: Supa taluk, Uttara Kannada, Karnataka, India(15.247023499763804, 74.38629523681011)
Comments: Tiger

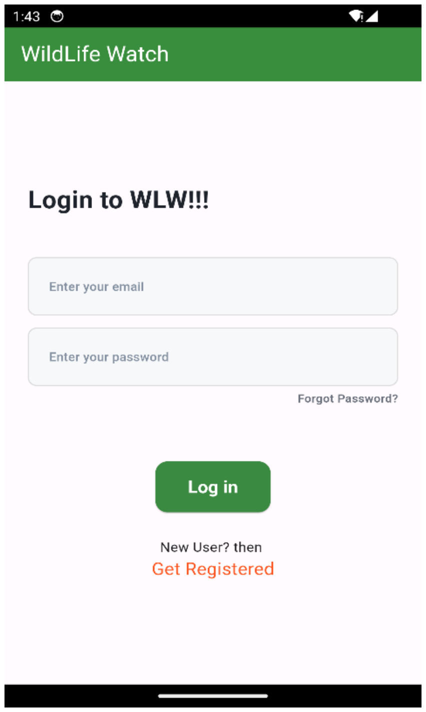
Rajan Gouda
Email: rajan@gmail.com
Date: 2024-04-16
Place: Mysuru, Mysuru taluk, Mysuru District, Karnataka, 570001, India(12.294820797950814, 76.6482975770743)
Comments: Elephant at Mysore

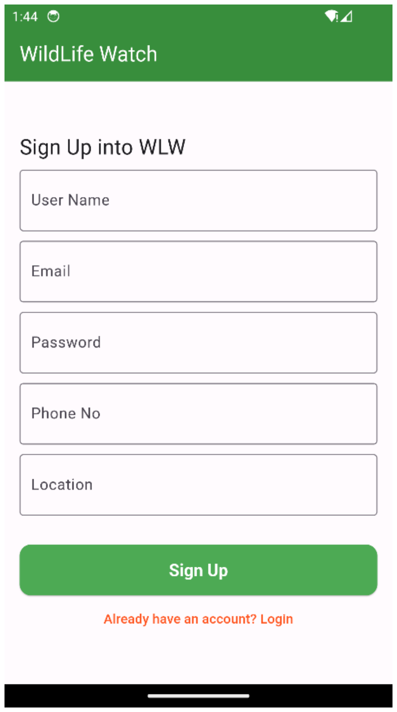
9.10 Animal Recognition

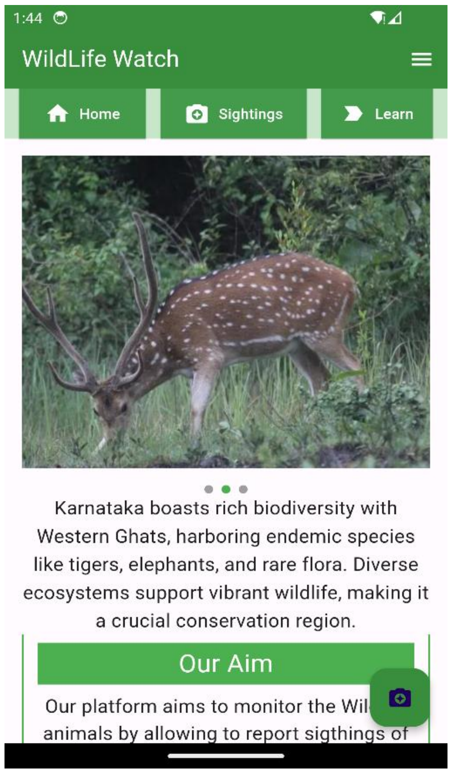


Identified Animal: green mamba

9.11 Application Interface

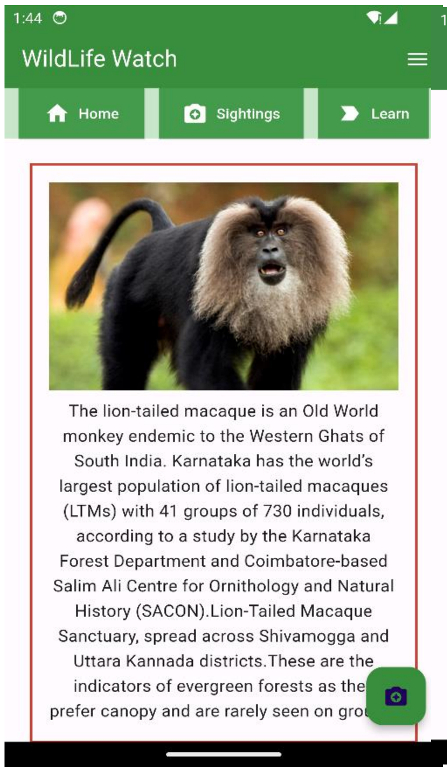




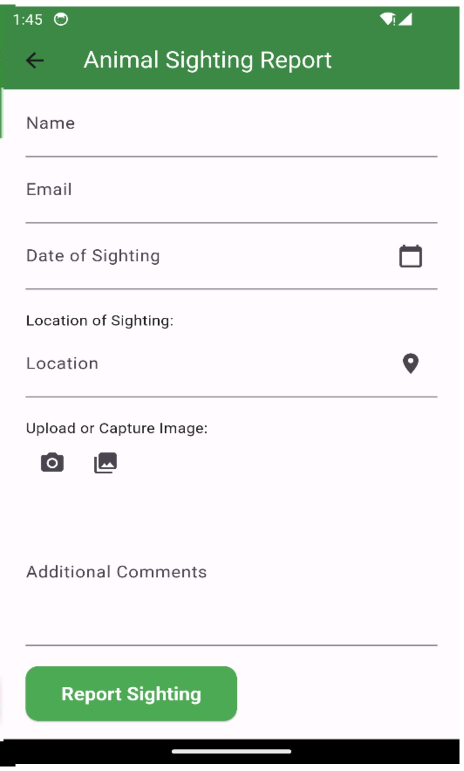


Karnataka boasts rich biodiversity with Western Ghats, harboring endemic species like tigers, elephants, and rare flora. Diverse ecosystems support vibrant wildlife, making it a crucial conservation region.

Our platform aims to monitor the Wil animals by allowing to report sightings of



The lion-tailed macaque is an Old World monkey endemic to the Western Ghats of South India. Karnataka has the world's largest population of lion-tailed macaques (LTMs) with 41 groups of 730 individuals, according to a study by the Karnataka Forest Department and Coimbatore-based Salim Ali Centre for Ornithology and Natural History (SACON).Lion-Tailed Macaque Sanctuary, spread across Shivamogga and Uttara Kannada districts.These are the indicators of evergreen forests as the prefer canopy and are rarely seen on grou



The App version of WildLife Watch contains same functions as their in website like login/signup system, learn section, animal sightings report etc.

CHAPTER 9
APPLICATION

1. Community Engagement and Citizen Science:
   Through "Wildlife Watch," local communities and citizen scientists can actively participate in wildlife monitoring by reporting sightings and sharing observations, contributing valuable data to scientific research and conservation efforts.

2. Tourism and Eco-Tourism Promotion:
   The platform supports responsible wildlife tourism and eco-tourism by providing information on areas with rich biodiversity, enabling tour operators and tourists to plan trips that minimize negative impacts on wildlife and their habitats.

3. Policy and Decision-Making Support:
   Policymakers and environmental agencies can utilize the comprehensive data collected by "Wildlife Watch" to make informed decisions regarding wildlife protection regulations, habitat preservation, and resource allocation, ensuring effective conservation policies.

4. Research and Data Analysis:
   Researchers benefit from access to extensive datasets on wildlife sightings and movements, supporting various scientific studies such as population dynamics, species distribution modeling, and the impacts of environmental changes on wildlife.

5. Emergency Response and Wildlife Rescue:
   In wildlife emergencies, "Wildlife Watch" provides real-time alerts and updates, aiding authorities and rescue teams in responding promptly to protect affected wildlife from natural disasters, poaching incidents.

6. Habitat Restoration and Management:
   Conservationists can utilize the data on wildlife movements and habitat use provided by "Wildlife Watch" to identify critical areas for habitat restoration projects, ensuring the sustainability of ecosystems and supporting wildlife populations.

CHAPTER 10

FUTURE SCOPE

The future scope of "Wildlife Watch" envisions leveraging cutting-edge technologies to enhance its capabilities in wildlife monitoring and conservation. One significant development is the integration of artificial intelligence (AI) and machine learning (ML) algorithms to analyze collected data more efficiently. These technologies can be employed to predict wildlife movement patterns, identify emerging threats to endangered species, and provide conservationists with actionable insights. By automating data analysis and developing predictive models, "Wildlife Watch" will be able to offer more accurate and timely recommendations, thus enabling more proactive and effective conservation efforts.

Another promising area for expansion is the incorporation of Internet of Things (IoT) devices for real-time wildlife monitoring. IoT-enabled cameras, tracking collars, and environmental sensors can continuously collect detailed data on wildlife behavior and habitat conditions. This real-time data can be integrated into the platfozrm, providing users with a comprehensive and up-to-date view of wildlife activities and environmental changes. The deployment of drones for aerial surveillance and data collection in remote or challenging terrains is also a potential enhancement, broadening the platform's monitoring capabilities. These advancements will significantly increase the granularity and immediacy of data available to researchers and conservationists.

The future scope also includes expanding the platform's reach and impact through strategic partnerships with global conservation organizations, academic institutions, and government agencies. By creating a network of contributors and stakeholders, "Wildlife Watch" can facilitate the exchange of knowledge, resources, and best practices in wildlife conservation. Multilingual support and region-specific features will make the platform more accessible to a global audience, ensuring that conservation efforts are inclusive and culturally sensitive. Additionally, enhancing mobile app capabilities and introducing offline functionality will increase user engagement and participation, especially in remote areas with limited internet access. These future developments aim to establish "Wildlife Watch" as a leading global tool for wildlife conservation, fostering a collaborative and informed approach to protecting biodiversity.

CHAPTER 11

CONCLUSION

In conclusion, "Wildlife Watch" represents a significant advancement in the realm of wildlife conservation and management. By using the power of technology, this innovative platform empowers individuals, communities, and organizations to actively participate in the protection of wildlife and their habitats. Through real-time reporting of wildlife sightings, access to educational resources, and data-driven insights, "Wildlife Watch" fosters a collaborative approach to conservation, where stakeholders from diverse backgrounds can contribute to a common goal.

The platform serves as a catalyst for raising awareness and promoting environmental stewardship on a global scale. By providing users with a deeper understanding of biodiversity, conservation challenges, and the importance of preserving natural ecosystems, "Wildlife Watch" inspires a sense of responsibility and commitment to sustainable practices. This heightened awareness not only benefits wildlife populations but also contributes to the broader movement towards a more harmonious relationship between humans and the natural world.

With ongoing advancements in technology and increasing global awareness of environmental issues, the platform is poised to evolve and expand its impact. Through continued innovation, strategic partnerships, and community engagement, "Wildlife Watch" will continue to play a pivotal role in shaping the future of wildlife conservation, paving the way for a more resilient and biodiverse planet for generations to come.

REFERENCES

[1] Soledad Luna, Margaret Gold, Alexandra Albert, Luigi Ceccaroni, “Developing Mobile Applications for Environmental and Biodiversity Citizen Science: Considerations and Recommendations”, (June 2018) Multimedia Tools and Applications for Environmental & Biodiversity Informatics. Web:https://povesham.wordpress.com/2018/08/08/developing-mobile-applications-for-environmental-and-biodiversity-citizen-science-considerations-and-recommendations

[2] Wright, M. D., Turner, W. C., & others, "A review of wildlife monitoring technologies”, (2016) Journal of Wildlife Management

[3] Eweoya, I. O., Ajayi, O. J., & others, "Design and Implementation of Web-based GIS for Wildlife Management System", (2017) Journal of Geographic Information System

[4] Shanahan, D. J., Fuller, R. A., & others, “The role of citizen science in wildlife monitoring and conservation”, (2015) Trends in Ecology & Evolution, 30(8), 462-470.

[5] Bhatia, N., & Saini, S, "Real-time web applications: Technologies and challenges", (2012) International Journal of Computer Applications

[6] Brenda McComb, "Monitoring Animal Populations and Their Habitats: A Practitioner's Guide",2010, CRC Press

[7] Michael L. Morrison, William M. Block, M. Dale Strickland, William L. Kendall "Wildlife Study Design", 2001, Springer

[8] Janis L. Dickinson, Benjamin Zuckerberg, and David N. Bonter, "Citizen Science: A Tool for Integrating Studies of Human and Natural Systems", 2010, Annual Review of Ecology, Evolution, and Systematics

[9] Krystal E. Culligan, Wesley E. Koenig, Kristina M. Williams, "Using Web-Based Citizen Science to Monitor Invasive Species: Problems and Solutions", 2019, Frontiers in Ecology and Evolution

[10] P. C. Joshi, "Ecology and Wildlife Conservation", 2016, Agrobios (India)