**Department of Computer Science and Engineering,**

**DAYANANDA SAGAR COLLEGE OF ENGINEERING**

*(An Autonomous Institution affiliated to VTU, Accredited with NAAC ‘A’ Grade)*

**BANGALORE – 560078**

A Mini-Project Report on

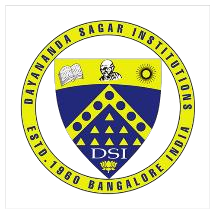
**Bird Species Encyclopedia**

Submitted in 5th Semester

**Bachelor of Engineering**

In

**Computer Science and Engineering**



Submitted by

##### Abhishek 1DS16CS130

##### Vadiraj K R 1DS16CS131

##### Ruthwik H P 1DS16CS125

##### Mukunda K Das 1DS16CS062

##### Mehul Poddar 1DS16CS058

##### Under the guidance of

##### Dr. Mohammed Tajuddin

##### Associate Professor

##### Dept. of CSE, DSCE

2018-2019

**Abstract**

In today’s world, there are multiple species of birds that are being recorded and since our world has reached the stage where everything can be found by a small device that fits into our pockets why not being able to find the information of an given bird species in our fingertips. A consolidated database which can be used from any location with a stable internet connection will be much better to use than having to search multiple sources to find one inquiry. This will reduce efforts of wildlife enthusiasts having to search multiple websites to find one query.

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Number** | **Topic** | **Page Number(s)** |
| 1 | Introduction | 4 |
| 1.1 | Objectives of the Project | 4 |
| 2 | Design/Implementation | 5-10 |
| 2.1.1 | Login Page | 5 |
| 2.1.2 | Insertion Page | 6 |
| 2.1.3 | Query Page | 6 |
| 2.1.4 | Database Design | 7 |
| 2.2 | Implementation | 8-10 |
| 3 | Testing and Analysis | 11 |
| 4 | Snapshots of the Results | 12-14 |
| 4.1.1 | How a query looks | 12 |
| 4.1.2 | Queried Entry | 13 |
| 4.1.3 | Conservation Status button | 14 |
| 5 | Conclusion and Future Enhancement | 15 |
| 5.1 | Conclusion | 15 |
| 5.2 | Future Enhancement | 15 |

1. **Introduction**

In today’s world, there are multiple species of animals, birds, etc, that are being recorded and since our world has reached the stage where everything can be found by a small device that fits into our pockets why not being able to find the information of an given bird species in our fingertips. A consolidated database which can be used from any location with a stable internet connection will be much better to use than having to search multiple sources to find one inquiry. This will reduce efforts of wildlife enthusiasts having to search multiple websites to find one query.

Firebase is a [mobile](https://en.wikipedia.org/wiki/Mobile_application) and [web application](https://en.wikipedia.org/wiki/Web_application) development platform. Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud.

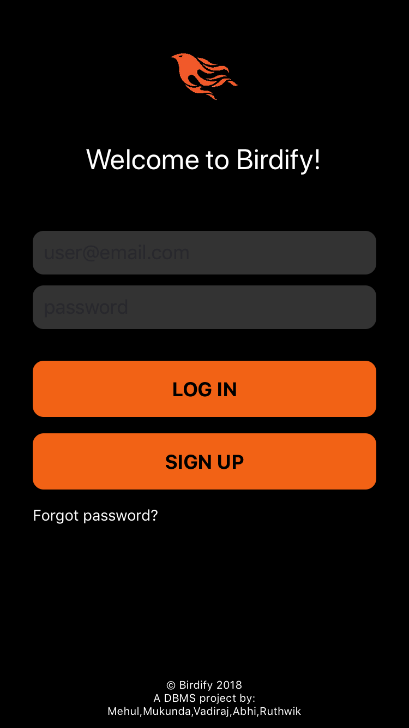
Our project titled “Bird Species Encyclopedia” uses Firebase and React Native (for the GUI). We are developing a NoSQL database that uses firebase as the database and React Native for the GUI. This project will highlight features of components used in Firebase.

* 1. **Objectives of the Project**
* Reduce the efforts of wildlife enthusiasts.
* Create a consolidated database.
* Use Firebase.

1. **Design/Implementation**

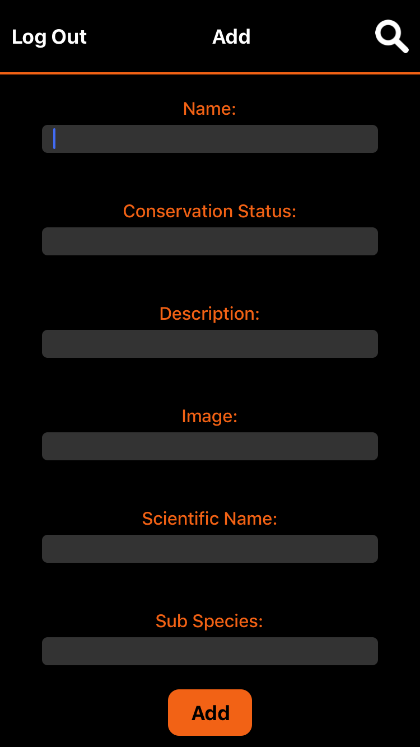
This project highlights the features of Firebase and React Native. This project demonstrates how a wildlife enthusiast can look up information about a certain bird using the app which is connected to the database.

* + 1. **Login Page**

****

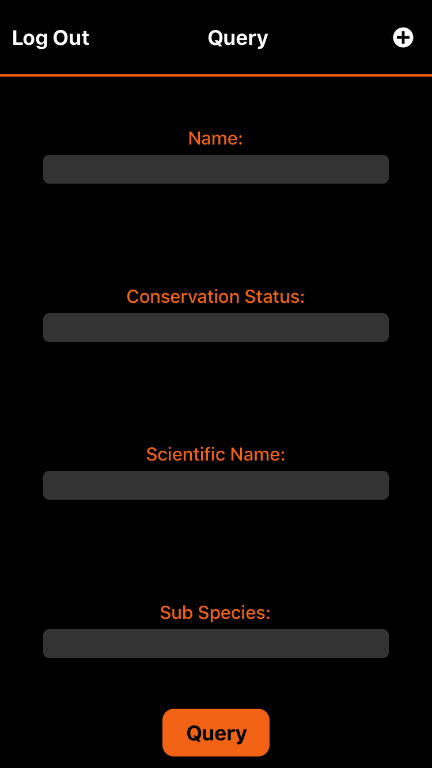
Here is the login page where we have created for the users to input their login details and it will be authenticated.

* + 1. **Insertion Page**

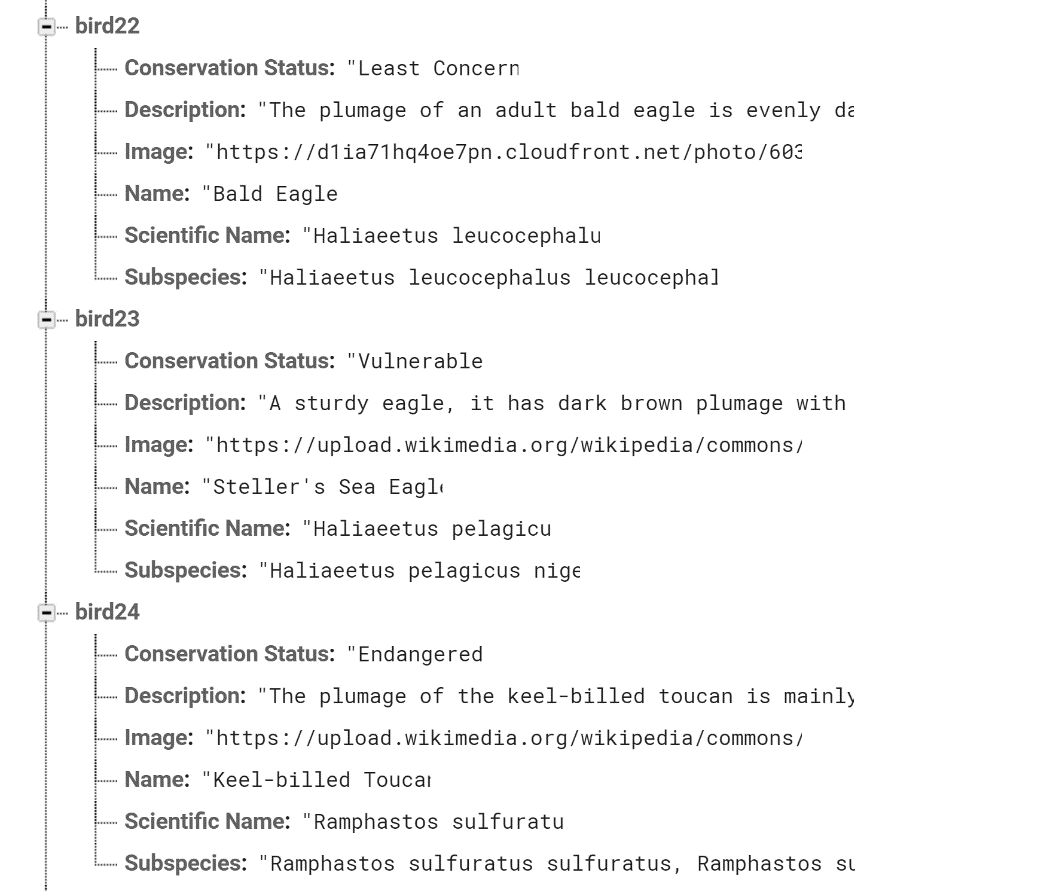


Here is what the user can insert data into the database after logging into the app.

* + 1. **Query Page**

Here is what the query page looks like. This is where the user after logging in can search for the birds depending on the parameters shown.

**2.1.4 Database Design**



Here is a sample of the data and how we have designed the database to be like. bird22 and such are the objects and Name, Image, Scientific Name are it’s attributes. Each bird has conservation status, description, image, name, scientific name and subspecies.

* 1. **Implementation**

1. React Native
2. Firebase

In this project, we have implemented the following tools. React Native for the GUI and Firebase for the database. React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android.

Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. Firebase synchronizes application state, and React Native re-renders the application UI based on state changes. The UI listens for any search parameters and searches for the query only after the button to query has been pressed. The UI, after listening for the button click, will search the database depending on the search parameters given by the user. This is done by calling the objects’ attributes. This will allow for the results to be displayed if the data exists for the query that has been given by the user.

**Code Snippet to add data to database:**

let data = this.state.birdData;

      const ctr = Object.keys(data).length + 1;

      let tempObj = {};

      tempObj['Conservation Status'] = this.state.conStat;

      tempObj['Description'] = this.state.desc;

      tempObj['Image'] = this.state.image;

      tempObj['Name'] = [this.state.name](http://this.state.name/);

      tempObj['Scientific Name'] = this.state.scientificName;

      tempObj['Subspecies'] = this.state.subSpecies;

      data['bird' + ctr] = tempObj;

      firebase.database().ref().update({ birds: data });

**Code Snippet for Query Matching:**

let data = []

      for(var i = 1; i <= Object.keys(this.state.birdData).length; i++) {

        const bird = this.state.birdData['bird' + i];

        let n = bird['Name'];

        let sn = bird['Scientific Name'];

        let ss = bird['Subspecies'];

        let cs = bird['Conservation Status'];

        if ([this.state.name](http://this.state.name/) !== '') { n = [this.state.name](http://this.state.name/); }

        if (this.state.scientificName !== '') { sn = this.state.scientificName; }

        if (this.state.subSpecies !== '') { ss = this.state.subSpecies; }

        if (this.state.conStat !== '') { cs = this.state.conStat; }

        if (n.toUpperCase() === bird['Name'].toUpperCase() &&

        sn.toUpperCase() === bird['Scientific Name'].toUpperCase() &&

        ss.toUpperCase() === bird['Subspecies'].toUpperCase() &&

        cs.toUpperCase() === (bird['Conservation Status'].toUpperCase())) {

          data.push(bird);

        }

      }

**Code Snippet to read data from database:**

let tempData = {}

    firebase.database().ref().child('birds').on('value', dataSnapshot => {

      dataSnapshot.forEach(childSnap => {

        let tempObj = {}

        childSnap.forEach(childParam => {

          tempObj[childParam.key] = childParam.val();

        });

        tempData[childSnap.key] = tempObj;

      });

    });

    this.setState({ birdData: tempData });

1. **Testing and Analysis**

The user must first login to the app where he/she will be authenticated by the app by checking the database for the login details.

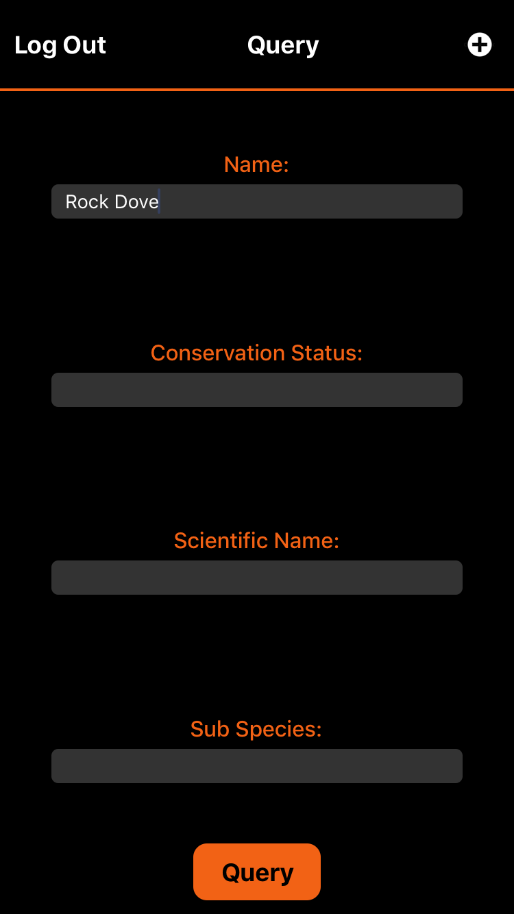
Once the user has logged in to the app, the user can either insert data into the database on the 6 attributes, Conservation Status, Description, Image, Name, Scientific Name and Subspecies. These 6 attributes are the ones that will be displayed when a query has been.

The other part of the user’s abilities will be to query data. The 6 attributes will be displayed as cards on the app. Each bird that fits the query will be displayed as cards.

The user can query data by accessing the query page. Then by adding search parameters, and clicking on the query button, the successful search results will be displayed as cards.

The user can add data by going to the insertion page by clicking on the plus sign and he/she will be directed to the insertion page where they will be asked to add data for the 6 attributes shown.

1. **Snapshots of the Results**
   * 1. **How a query looks**



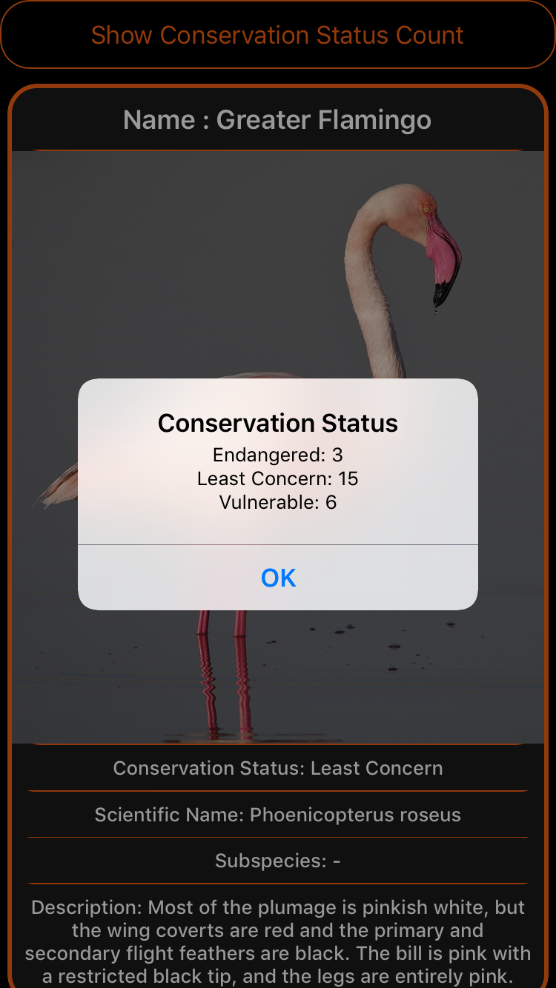
The user will be able to enter anything in the various fields as shown above and by pressing on the “Query” button, the app will search the database for whichever field has been entered.

* + 1. **Queried entry**

****

This is what a successful search looks like. The following details will be shown to the user depending on the search parameters that has been entered.

* + 1. **Conservation Status button**

****

The button on the top of the interface, “Show Conservation Status Count”, is a button that will display how many birds that are in the database have what conservation status assigned to them.

1. **Conclusion and Future Enhancement**
   1. **Conclusion:**

In conclusion, we have seen that, by using Firebase, we have successfully allowed wildlife enthusiasts to search about any bird species that they may have seen and would like to know more about. Firebase allowed for all the data to be stored as objects and hence easier and faster access as well as making the data easier to search for as well.

* 1. **Future Enhancement:**

As nothing is completely finished, this project can be further developed. For the moment, we have used a very limited amount of data when compared to the actual number of bird species that exist today.

The project can be enhanced further by making the GUI more user-friendly and by adding more descriptors about the birds such as wingspan or height directly into the database. We could also add more data that would allow for a more complete encyclopedia of the world’s bird species.