Zap App – A mobile-optimised progressive web application to record Pangolin sightings

CI609 – Advanced Web   
Application Development

8th November 2022  
  
Jazer Barclay  
(20837308)

**Table of Contents**

[1. Introduction 2](#_Toc118801992)

[2. Requirements Analysis 3](#_Toc118801993)

[3. System Architecture 4](#_Toc118801994)

[4. Technical Research and Development 5](#_Toc118801995)

[4.1. Tooling 5](#_Toc118801996)

[4.2. Frontend 5](#_Toc118801997)

[4.3. Backend 5](#_Toc118801998)

[4.4. Database 6](#_Toc118801999)

[5. Reflection 7](#_Toc118802000)

[6. References 8](#_Toc118802001)

[7. Appendix 9](#_Toc118802002)

Introduction

* Outline problem

A research team studying threats to Pangolins in the wild commissions you to develop a

mobile-optimised progressive web application to record Pangolin sightings and mortalities.

Pangolins are the world’s most trafficked mammal and are threatened solely by human

impacts. In addition to tracking and poaching, many pangolins in Africa face the risk of being

killed on roads and by electric fences. The app should enable game rangers and local

communities across Southern Africa to log pangolin sightings and mortalities, enabling the

research team to study these threats.

This app should allow users to take a picture or select one from the gallery and select

whether they have seen the Pangolin alive or dead. If dead, users should select the type of

mortality (fence death: electrocution; fence death: caught on non-electrified fence; road

death; other). Users also should have an option to add additional notes, such as fence or

road type. The app should automatically pick up the user's location and submit it together

with the recorded information.

As user may be in remote areas with no signal when they encounter pangolins, the app

should provide key functionality offline. It should be able to store information locally and

upload that information later when they have an internet connection.

A client-server solution involving a REST API is preferred to allow other organisations to

develop their own client applications against the system.

Related links:

https://developer.mozilla.org/en-US/docs/Web/API/Geolocation\_API

https://developers.google.com/web/fundamentals/media/capturing-images

https://developer.mozilla.org/en-US/docs/Web/API/Navigator/Online\_and\_offline\_events

* Short description of the application
* What problem does it address

Requirements Analysis

* How requirements were identified
* Which requirements were identified
* Requirements prioritisation

System Architecture

Overview diagram (data flows, protocols/methods)

REST API

* Design considerations
* Summary of endpoints, methods and data (full specification in separate API Documentation)

Any additional web APIs used

* Research into suitable APIs (how do they compare?)
* Any API experiments carried out
* Chosen API(s) (why this/these ones?)

Technical Research and Development

* 1. Tooling
* Tools and plugins researched (how do they compare?)
* Any tooling experiments carried out
* Chosen tools/plugins (why these ones?)
* Development/build setup
  1. Frontend
* Frameworks/technologies researched (how do they compare?)
* Any code experiments carried out
* Chosen framework/technology (why this one?)
* Frontend implementation
  + Technical design
  + UI design (responsive)
* Evaluation
  + Testing
  + standards compliance
  + accessibility
  1. Backend
* Frameworks/technologies researched (how do they compare?)
* Any experiments carried out
* Chosen framework/technology (why this one?)
* Backend implementation
  + Technical design
  + Data validation
* Evaluation
  + Testing
  1. Database
* Databases and APIs researched (how do they compare?)
* Chosen database/API (why this one?)
* Database design
  + Structure / schema
  + Field names and data types

Reflection

* Reflection on tooling and technology choices
* Strengths and weaknesses of the application
* Possible improvements and further development

References

* Use Harvard referencing style!
* Only include sources actually referenced in the text!

Appendix

* Any supporting materials