ECOSIA - "TREE SHOWCASE"

TDD

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Responsibilities** | **Name – Function** | **Department - Company** | **Date** | **Signature** |
| Written by | Dhirendra Singh | SopraSteria | 07-Dec-2020 |  |
|  |  |  |  |  |

All rights reserved. **Confidential and proprietary document**.

© **EADS** 2020. All rights reserved. **Confidential and proprietary document**.

**TABLE OF CONTENT**

[1. Introduction 3](#_Toc58258552)

[1.1. Purpose of the document 3](#_Toc58258553)

[1.2. Scope of the document 3](#_Toc58258554)

[2. Wireframe 4](#_Toc58258555)

[2.1. Project Setup 4](#_Toc58258556)

[2.2. Configuring CSS 4](#_Toc58258557)

[2.3. Code Implementation: 5](#_Toc58258558)

[2.4. Functional View 5](#_Toc58258559)

[3. APP View: 6](#_Toc58258560)

[3.1. In Large Size Viewport View: (Desktop, Laptop, Large Screen) 6](#_Toc58258561)

[3.2. In Medium Size Viewport View: (Ipad view, Landscape view) 6](#_Toc58258562)

[3.3. In Small Size Viewport View: (mobile) 8](#_Toc58258563)

# Introduction

## Purpose of the document

The Architecture Dossier presents the general and detailed architecture of the target solution for the Frontend Developer Challenge.

This document is initiated during the Inception phase for Agile. At this point, the different architecture candidates have been described and compared. The Architecture Dossier is completed during the Go-Live Review in Agile.

The solution to be produced is described in a **complementary manner** by one dossiers, The Architecture Dossier:

* The **Architecture Dossier** formalizes the architecture description of the target solution. It includes the System Requirements that prescribe the architecture at the system level.

In particular, the software specification dossier must attempt to describe the components and the interfaces identified in the architecture dossier. The intent of this document is to be submitted for approval. The Architecture Dossier (TDD) contains System Requirements for the project.

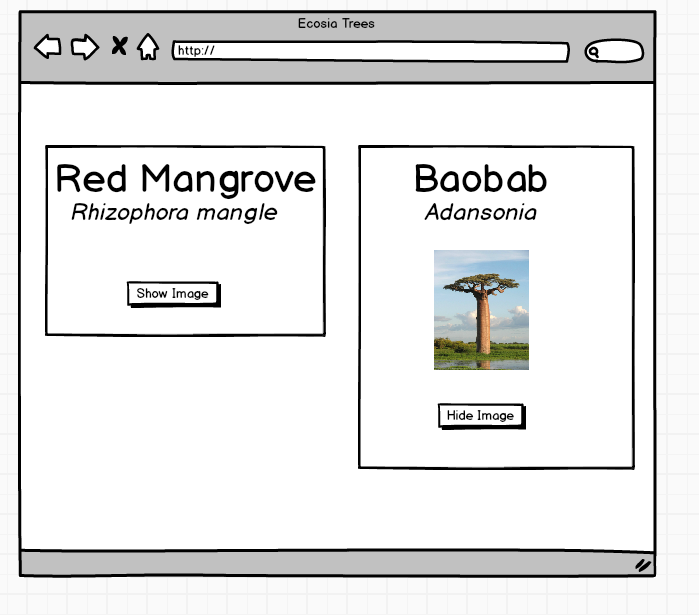
## Scope of the document

This Architecture Dossier documents the detailed specification for the Frontend Developer Challenge solution to be submitted across the HR department for selection process.

This document only tackles the requirement of as per User Stories

|  |  |  |
| --- | --- | --- |
| EPIC | USER STORY | ACCEPTANCE CRITERIA |
| As a user, I need to see the list of trees. | As a user, I see list of trees in App fetched from given API. | Ensure the user is able to:  \*see the list of tree items. \*Api should be called. *if no data the app displays the “Loading…” text* |
|  | As a user, I can the toggle button in every tree item. | Ensure the user is able to:  \*click on button  There should be button on every single item. |
|  | As a user, I can click and toggle HIDE/SHOW item. | Ensure the user is able to:  \*toggle the button  *on toggling the image will hide and show accordingly.* |

# Wireframe



## Project Setup

First, we are going to create a React project using Create React App as follows:

*$ yarn create react-app json-request*

*$ cd json-request*

*$ yarn start*

You will see a new browser tab open at the URL http://localhost:3000. You can stop the running React app using Ctrl+C.

If you have full package then you can simple run below command:

*$ yarn install*

## Configuring CSS

We are going to write style in bootstrap and a basic CSS reset. You can install “BOOTSTRAP” by using below command:

*$ yarn add bootstrap*

And paste below line in “index.js” file, show that bootstrap will apply globally (in application level)

import "bootstrap/dist/css/bootstrap.css";

## Code Implementation:

Please see the attached zip source code of tree-showcase application.

## Functional View

* Programmatically fetch the list of trees (in JSON format) from https://s3.eu-central-1.amazonaws.com/ecosia-frontend-developer/trees.json
* Display a list of trees including the species name and image (hidden by default), using the provided layout sketch
* Allow the showing/hiding of the tree image
* Include unit/component testing
* Here's a sample of the JSON data returned by the endpoint above. You can use this to get an idea of the data returned before beginning your challenge, but please remember that your application should fetch this data from the live endpoint listed above

*{*

*"trees": [*

*{*

*"name": "Baobab",*

*"species\_name": "Adansonia",*

*"image": "https://upload.wikimedia.org/wikipedia/commons/3/36/Baobab\_Adansonia\_digitata.jpg"*

*},*

*…….*

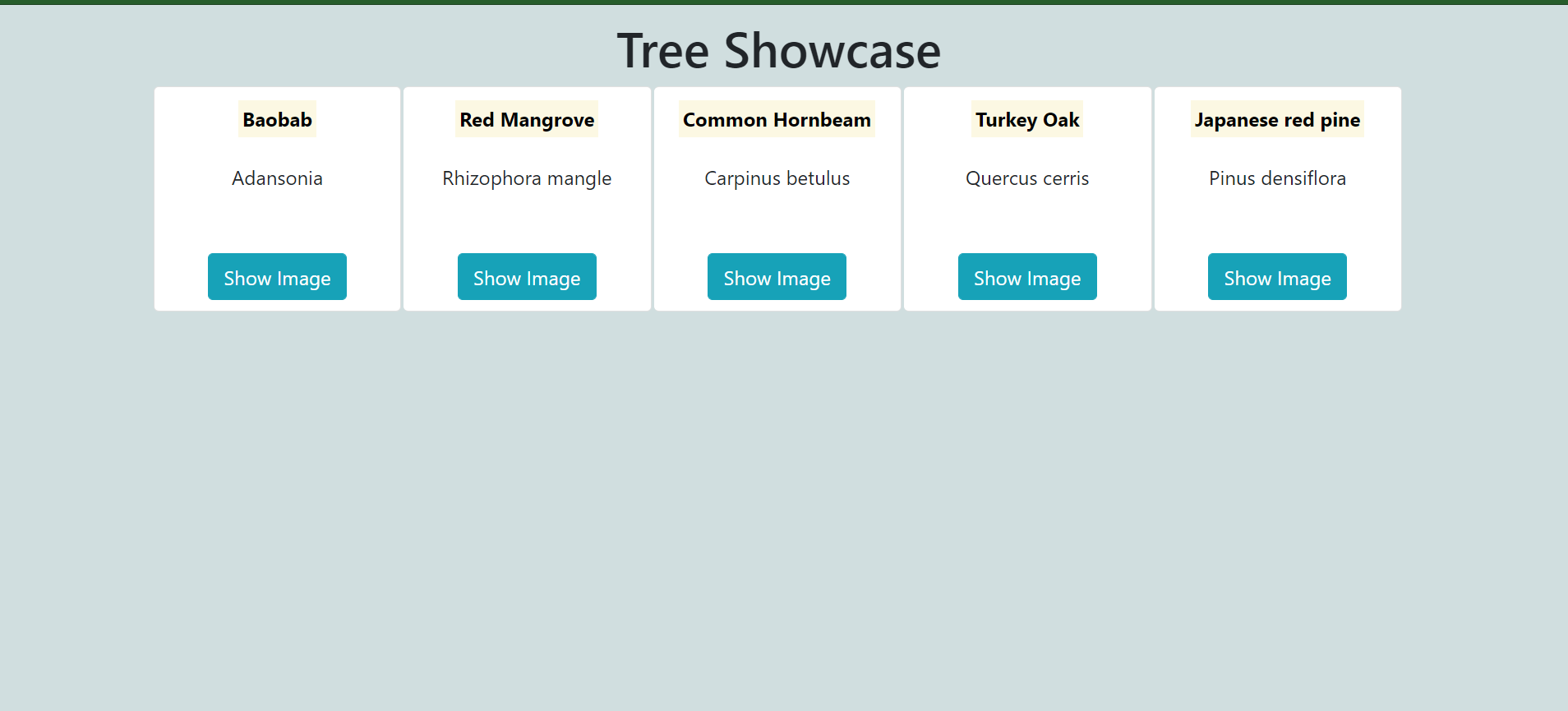
*]*

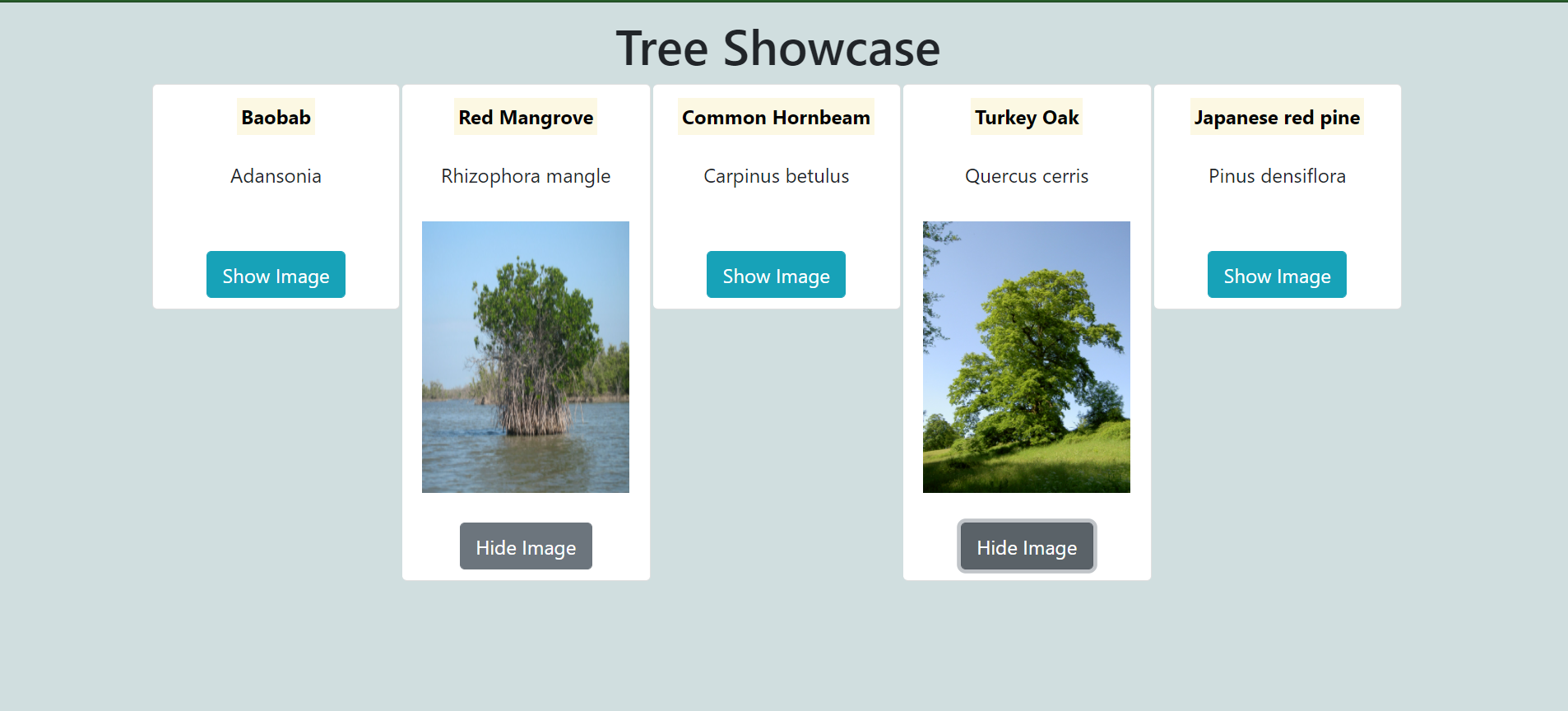
*}*

# APP View:

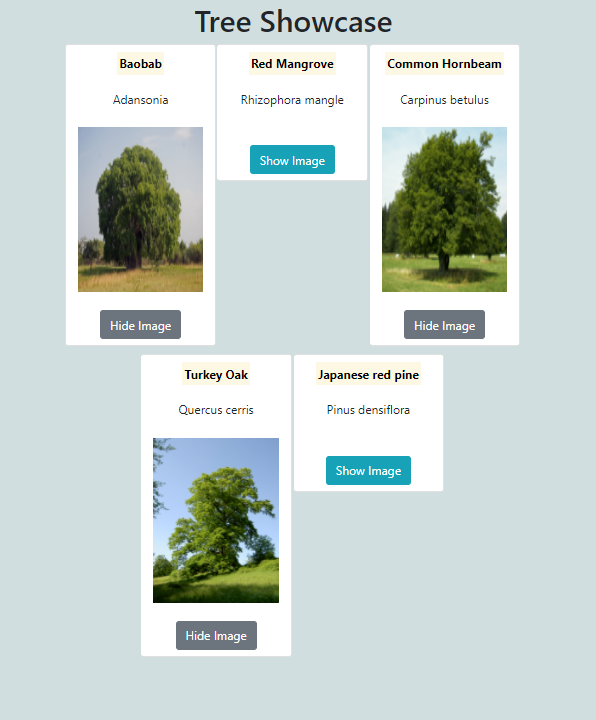
Below are the some snapshot for the application in different layouts:

## In Large Size Viewport View: (Desktop, Laptop, Large Screen)





## In Medium Size Viewport View: (Ipad view, Landscape view)



## In Small Size Viewport View: (mobile)

