

# **Software Implementation and Testing Document**

**For  
Group 4**

Version 1.0

**Authors:**  
Jordan Rydzinski  
Parker Lee  
Bao Tran  
Ingrid Perez-Valezquez  
Aidan Mahoney

## **1. Programming Languages (5 points)**

*List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).*

- Python: Used for back-end logic and server-side operations. We chose Python because of its extensive support for APIs and data management.
- JavaScript/TypeScript: Used for front-end interactivity, specifically the logic for the Species Guessing Game (e.g., the unblurring effect and score calculation) and the Geoguessr game.
- HTML5/CSS3: Used to structure the web application and style the user interface. CSS is specifically used for the visual "blur" transitions and the responsive layout of the game panels.

## **2. Platforms, APIs, Databases, and other technologies used (5 points)**

*List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).*

- GitHub: Used as our version control platform to host source code, track issues, and manage documentation.
- Supabase: Utilized as the back-end database for user registration and saving high scores on the leaderboard.
- Explore.org & Smithsonian National Zoo APIs: Used to pull live animal camera feeds and data for the "Random Zoo Cam" feature.
- Google Maps API & Leaflet.js: These are used for the Geoguessr game to display the interactive map and calculate the distance between the user's pin and the animal's origin.
- Beautiful Soup: A Python library used for web scraping to gather additional animal facts or data where formal APIs are unavailable.
- React: Used as the front-end framework to build a dynamic and responsive user experience.
- Digital Ocean: Used to create a server to host the application and handle recurring API calls.
- Namecheap: Used for registering domain and DNS for web server.

## **3. Execution-based Functional Testing (10 points)**

*Describe how/if you performed functional testing for your project (i.e., tested for the **functional requirements** listed in your RD).*

## **4. Execution-based Non-Functional Testing (10 points)**

*Describe how/if you performed non-functional testing for your project (i.e., tested for the **non-functional requirements** listed in your RD).*

## **5. Non-Execution-based Testing (10 points)**

*Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).*