# User Documentation – Project Setup & Maintenance Guide

## Overview

This document serves as a guide for developers and maintainers of the project. It explains the project structure, setup instructions, and troubleshooting tips to assist new team members in running and extending the project smoothly.

## Project Structure Overview

* /src: Contains all source code (backend/frontend logic).
* /public: Static files used in the UI (images, CSS, etc.).
* /docs: Project documentation and reports.
* README.md: Introductory file with project summary and setup info.
* env.example: Template for environment configuration.
* package.json / pom.xml: Dependency configuration for Node.js/Java/Maven-based projects.
* config/: Database and app configuration files.
* scripts/: Utility scripts for setup or data migration.

**Note:** All images should be at least **1920x1080 px** to ensure quality in the UI.

## Initial Setup – How to Run the Project

**1. Clone the Project:**  
 git clone https://github.com/SBNA-Game-Show/Geo-Guesser.git  
 cd your-project-name

**Or**  go to vs code and cmd+Shift +P opens the palette click git clone

**2. Install Dependencies:**

**Backend**

* cd backend
* npm install
* npm start

**Frontend**

* cd frontend
* npm install
* npm install leaflet react-leaflet
* npm run dev  
    
  **Tip:** Open **two terminal windows** in VS Code—one for backend and one for frontend—for smooth development.

**3. Environment Configuration(optional):**  
 Copy the .env.example file and rename it to .env.  
 Update all required fields such as DB\_HOST, DB\_USER, DB\_PASSWORD, PORT, etc.

## Testing the Application // in progress

* **To run unit tests:**  
   npm test   
  Or, for Maven:  
   mvn test

## Diagnostic Steps for Common Issues

## App not starting (port in use): Check for other instances on the same port and stop them.

## Database connection failed: Verify .env DB credentials and ensure DB server is up.

## Module not found errors: Run npm install or mvn clean install again.

## CORS or 404 errors in frontend: Confirm backend URL and CORS headers are properly configured.

## Game-Specific Improvements

* **Game Start Page:** Split into multiple pages for cleaner and more maintainable code.
* **Authentication & Security:** Implement enhanced login/auth measures.
* **Difficulty Levels:** Enable different game levels (e.g., beginner, intermediate, advanced) using landmarksData.js.
* **Images:** Ensure all landmark images are at least 1920x1080 px.
* **Code Optimization:** Refactor backend and frontend for readability, maintainability, and performance.

## Adding New Data or Upgrading Database

 **Add Data to DB:**

* Create a migration script or run a direct SQL/NoSQL insert query.

 **Integrate with App Logic:**

* Update relevant controller/service files where data is retrieved or processed.

 **Test Endpoints/UI:**

* Confirm frontend reflects new locations or game logic changes.
* Test using Postman or the browser.

## Contributing Notes

- Always create a new branch for each feature or bug fix.  
- Submit pull requests for review before merging to main.  
- Follow consistent commit messages and naming conventions.

## Checklist for New Developers

* Clone the repo and install dependencies.
* Configure .env file.
* Set up the database and seed data.
* Start development servers (backend & frontend) and test.
* Review README.md and this document for additional clarity.
* Ensure images are 1920x1080 px for optimal display.
* Verify game levels (difficulty) are working.

**Notes:**

* + Open two terminals in VS Code: one for backend and one for frontend.
  + Ensure MongoDB is running locally or connected to your DB server.
  + Backend communicates with the database; frontend communicates with backend API.
  + Frontend uses Leaflet for map rendering.
  + Game logic and leaderboard are served through backend API routes.