**User Guide: Setting Up and Initializing the Project Repository**

**Overview**

This user guide provides step-by-step instructions to set up the environment for your project, initialize a Git repository, and map it to a remote GitHub repository. These steps are essential for creating a dynamic version of your st\_shadow\_dynamic\_version project.

**Environment Setup**

**1. Install Required Tools**

1. **Install .NET SDK**:
   * Download the latest .NET SDK from the [official .NET website](https://dotnet.microsoft.com/).
   * Verify the installation:
   * dotnet --version
2. **Install Node.js**:
   * Download Node.js from the [official Node.js website](https://nodejs.org/).
   * Verify the installation:
   * node -v
   * npm -v
3. **Install Docker**:
   * Download Docker Desktop from the [Docker website](https://www.docker.com/).
   * Verify Docker installation:
   * docker --version

**2. Set Up Development Environment**

1. **Install IDEs/Editors**:
   * Use Visual Studio (for .NET backend) and Visual Studio Code (for React frontend).
2. **Create Project Folders**:
   * Backend:
   * mkdir backend
   * cd backend
   * dotnet new webapi -n MyBackend
   * Frontend:
   * mkdir frontend
   * cd frontend
   * npx create-react-app my-frontend

**3. Test Local Setup**

* Start the .NET backend:
  + dotnet run
* Start the React frontend:
  + npm start

**Initialize Local Git Repository and Map to Remote**

**Step 1: Open Terminal in the Project Folder**

Navigate to the folder containing your project:

cd /path/to/your/local/st\_shadow\_dynamic\_version

**Step 2: Initialize Local Git Repository**

Run the following command to initialize Git:

git init

**Step 3: Stage All Files**

Add all project files to the Git staging area:

git add .

**Step 4: Commit the Files**

Create an initial commit with a message:

git commit -m "Initial commit"

**Step 5: Map to Remote Repository**

Link your local repository to the GitHub repository:

git remote add origin https://github.com/OluOrija/st\_shadow\_dynamic\_version.git

**Step 6: Push to GitHub**

Ensure the branch is named main:

git branch -M main

Push the changes to the remote repository:

git push -u origin main

**Step 7: Verify Setup**

1. Check repository status:
2. git status
3. Confirm remote URL:
4. git remote -v

**Testing and Deployment**

1. **Run Docker Compose**:
   * Create a docker-compose.yaml file with the backend and frontend configuration.
   * Run the Docker containers locally:
   * docker-compose up
2. **Access Your Applications**:
   * Backend API: http://localhost:5000
   * Frontend App: http://localhost:3000

**Conclusion**

You have successfully set up the environment, initialized the Git repository, and mapped it to your GitHub remote repository. Your project is now ready for further development and deployment.