PREAPARED BY :
DevDynasty

Deployment Manual

INDEX

Introduction:

This manual provides comprehensive guidance on deploying both the frontend and backend components of the MoodSphere application, an emotion-based music recommender system. In this document, we cover system requirements, software requirements, installation procedures, and the deployment process for the application. Detailed steps are provided to ensure even those new to such deployments can successfully deploy and maintain the application.

Overview of Subtopics:

***System Requirements*: This section outlines the hardware and software prerequisites for running the MoodSphere application efficiently.**

Software Requirements and Installation Guides: Detailed instructions for installing all necessary software, including Git, Node.js, npm, and Vercel, as well as setting up your development environment on both Windows and macOS systems.

***Cloning the Repository*: Step-by-step process to clone the project repository from GitHub to ensure you have the necessary files on your local machine.**

***Setting Up the Project*: Instructions on how to prepare both the frontend and backend environments, including installing dependencies.**

***Running the Application Locally*: Explains how to start both the frontend and backend servers locally to test the application before deployment.**

***Deploying to Vercel*: A guide to deploying the frontend to Vercel, including how to connect your GitHub repository and manage deployment settings.**

***Updating and Redeploying*: Procedures for updating the application and redeploying to ensure the deployed application remains up-to-date with the latest changes.**

Frontend Deployment

System Requirements

- **Operating System**: Windows or macOS
- **Hardware**: Compatible with nearly all GPUs, ensuring smooth execution of the application.

Software:

- **Git**
- **Visual Studio Code (recommended IDE)**
- **Node.js**
- **npm (Node Package Manager)**
- **Vercel account for deployment**

Installation Guides

Before starting the deployment process, ensure you have the necessary tools installed on your system. Below are the guides for installing Git, Visual Studio Code, Node.js, npm, and creating a Vercel account on Windows and macOS.

Git:

- **Windows**: Download and install from Git for Windows[https://gitforwindows.org/]
- **macOS**: Install via Homebrew with brew install git, or download from Git SCM [https://git-scm.com/download/mac]

Visual Studio Code:

- **Windows & macOS**: Download and install from Visual Studio Code[https://code.visualstudio.com/Download]

Node.js and npm:

- **Windows & macOS**: Download and install from Node.js official site. npm is included with Node.js. [https://nodejs.org/en]

Vercel:

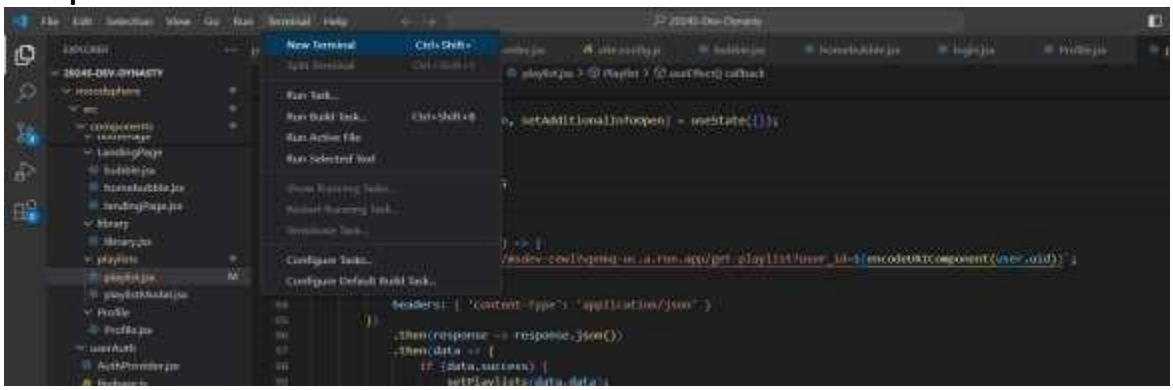
- Create an account at Vercel to prepare for deployment. [https://vercel.com/]

Step 1: Clone the Repository

- **Open your web browser and go to the GitHub repository at https://github.com/htmw/2024S-Dev-Dynasty**
- **Locate the green "Code" button on the GitHub page and click it.**
- **Copy the URL under "Clone with HTTPS".**
- **Open a terminal on your computer.**
- **Choose or create a directory on your local machine where you want to clone the repository.**
- **Type git clone, then paste the URL you copied. It should look something like this:**
- **git clone https://github.com/htmw/2024S-Dev-Dynasty.git**
- **Press Enter. Your local clone will be created.**

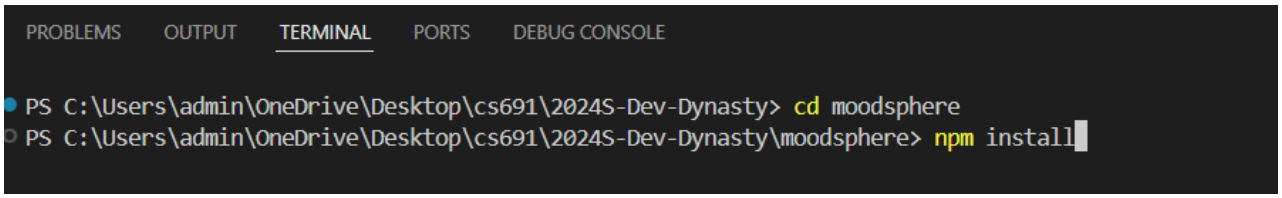
Step 2: Set Up the Project

- **Once the cloning is complete, navigate to the cloned directory by typing:**
- **cd 2024S-Dev-Dynasty**
- **open new terminal.**



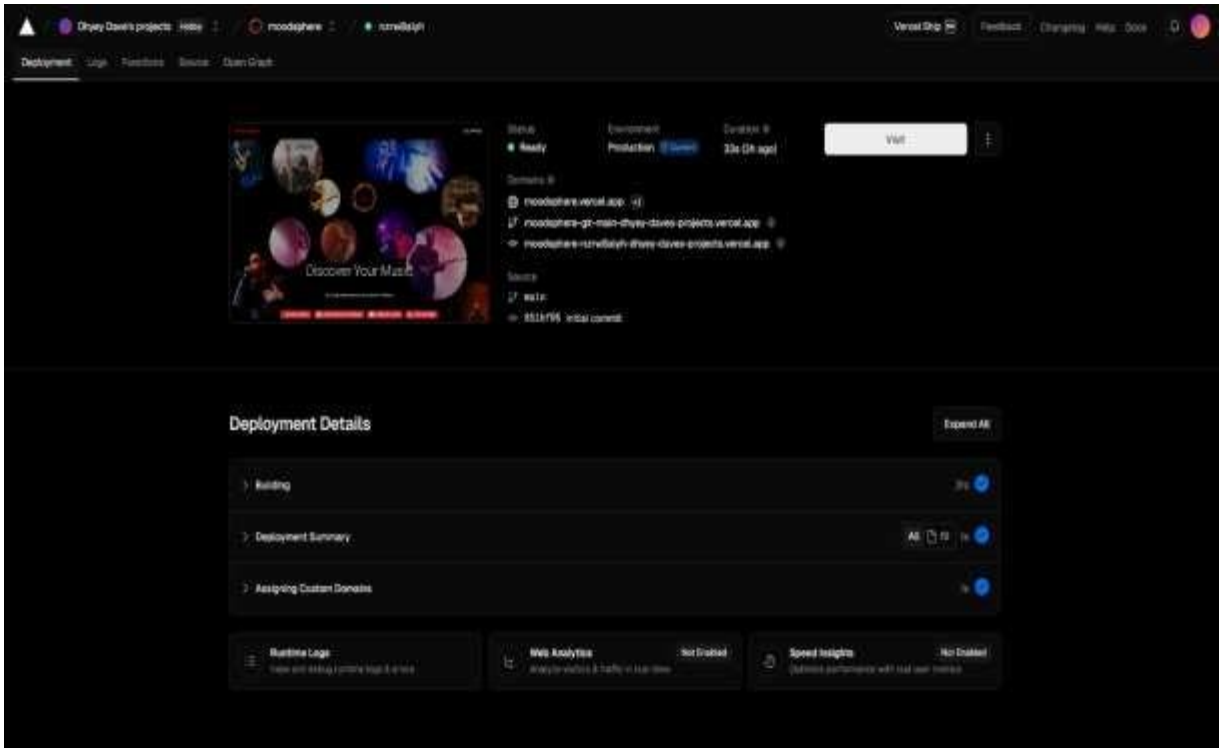
- **If your project has multiple subdirectories and the one you need to deploy is 'moodsphere', enter:**
- **cd moodsphere**
- **Install all the required npm packages by running:**

- **npm install**



Step 3: Run the Application Locally And Deploy to Vercel

- **Start the development server by typing:**
- **npm run dev**
- **This command starts the local server, typically accessible via http://localhost:3000 in your web browser. You can now view and test your application locally.**
- **Step 4: Deploy to Vercel**
- **Log in to your Vercel account or create a new one at https://vercel.app**
- **Once logged in, click on "New Project".**
- **You can either import your project directly from GitHub through Vercel or push your local repository to GitHub first and then import it. To connect to your GitHub repository, follow the prompts on Vercel to import your GitHub project.**
- **Follow the on-screen instructions to configure your project settings. Vercel will automatically detect settings based on the project type.**
- **After configuring, click on "Deploy". Vercel will build and deploy your application.**
- **Once the deployment is complete, Vercel will provide you with a public URL where your deployed site is accessible.**



Step 5: Update and Redeploy

- **Any updates you make locally can be pushed to your GitHub repository using:**
- **git add .**
- **git commit -m "Your update message."**
- **git push origin main.**
- **Vercel automatically redeploys your project with each push to the connected repository, keeping your deployed site up to date.**
- **Congratulations! You have successfully deployed your front-end application using GitHub and Vercel.**

Backend Deployment

Requirements:

- **A Google Cloud account[https://cloud.google.com/?hl=en]**
- **Docker installed on your machine [https://www.docker.com/]**
- **Git installed on your machine[https://git-scm.com/downloads]**
- **Access to the Docker repository "moodsphere/msdev"**

Step 1: Clone the Repository

- **Open a terminal on your computer.**
- **Navigate to the directory where you want to clone the repository.**
- **Clone the repository from GitHub by running:**
- **git clone https://github.com/htmw/2024S-Dev-Dynasty.git**

Navigate to the backend folder:

- **cd moodsphere-backend**

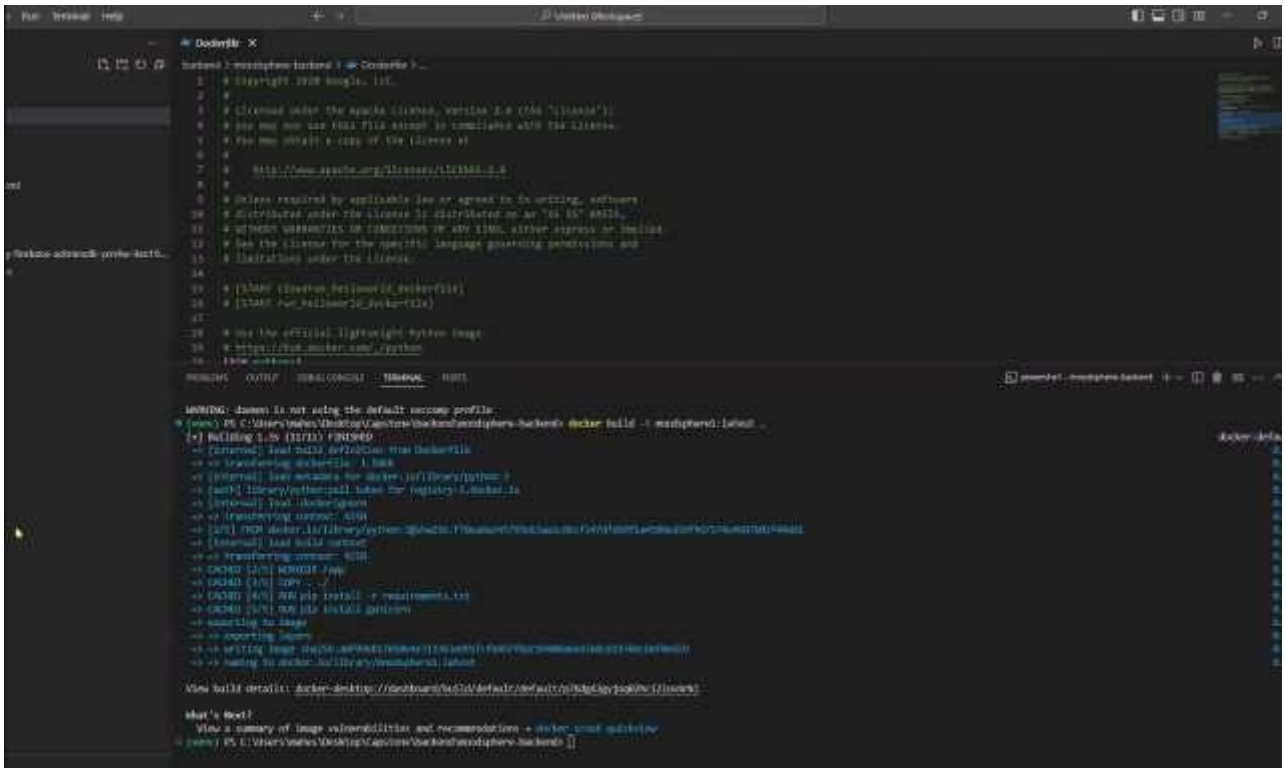
Step 2: Set Up Docker

- **Ensure Docker is installed and running on your machine. If not installed, download and install from Docker's official website.**
- **Log in to Docker if required using:**
- **docker login (To create username)**

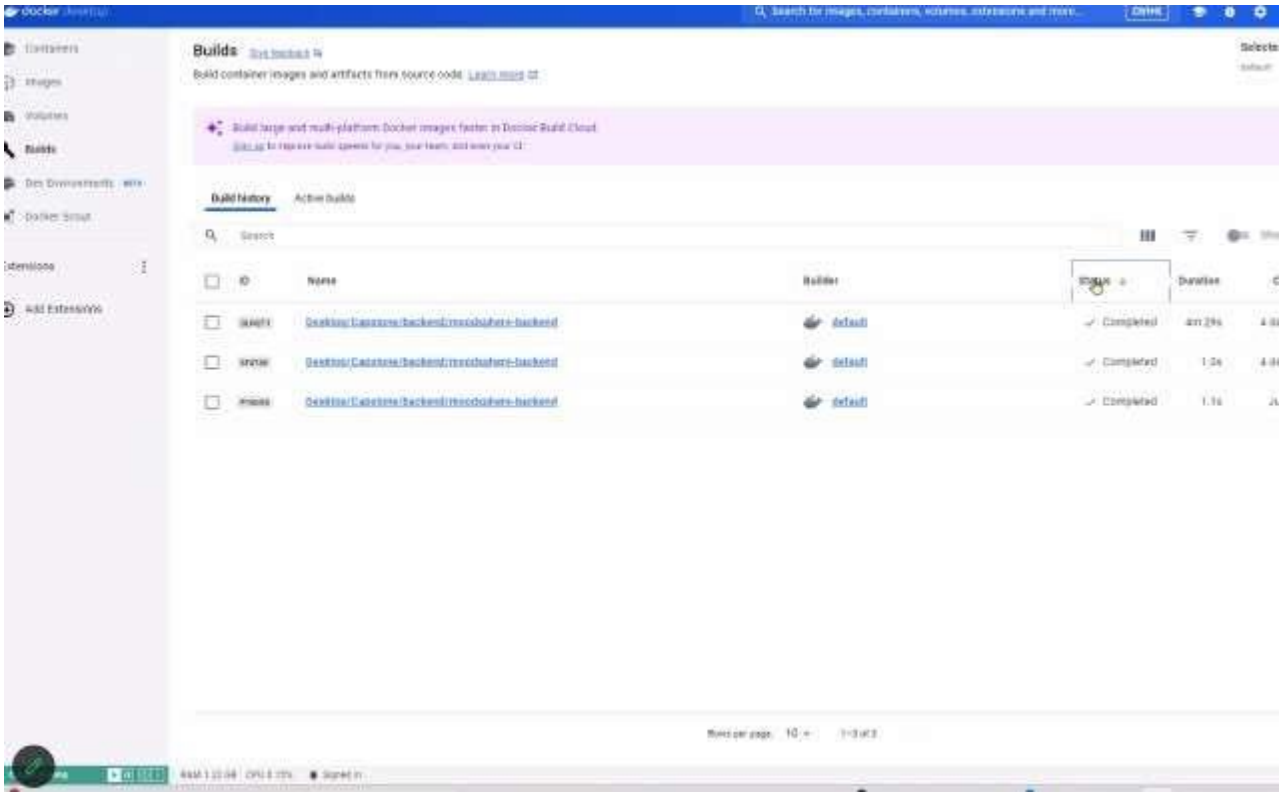
Step 3: Create the Docker Image

- **Start by creating a Dockerfile in the root of your project directory.**
- **Open your terminal or command line interface.**
- **Navigate to the directory that contains your Dockerfile.**
- **Run the following command to build the Docker image:**

  docker build -t your-app-name .



Step 4: Open Docker Desktop

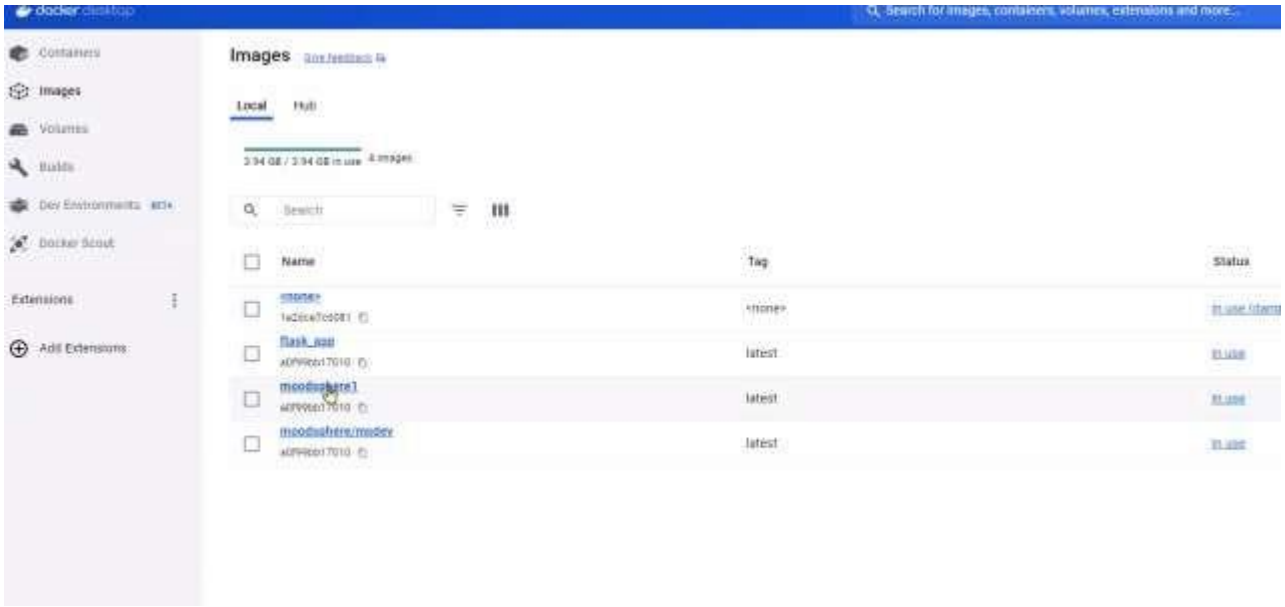


- **Launch Docker Desktop on your machine. It should automatically load the Dashboard where you can see all Docker components such as containers, images, and volumes.**
- **Navigate to the Images Section**

- **From the Docker Desktop sidebar, select Images. This section will display all Docker images available on your machine.**
- **Identify Your Project's Image**
- **In the Images tab, locate the image you just built. You can identify it by the name you tagged it with during the build process (e.g., your-app-name).**
- **Click on the image name to view more details.**
-

Step 5: Check Image Status

- **Look for the Status column next to your image. If the image is being used by a running container, it should display In use.**
- **Clicking on In use might further detail how many containers are running from this image and their statuses.**



Step 6: Go to the Containers Menu

- **From the Docker Desktop sidebar, click on Containers to see a list of all active and inactive containers.**
- **Identify and Manage Your Container**
- **In the Containers list, find the container running from your image. You can identify it by the image name (your-app-name).**
- **Check the Status to ensure it is running. You will also see the ports mapped (e.g., 8000:80), which tells you that the container is accessible through port 8000 on your host machine.**

Step 7: Test the Application

- **Open a web browser and navigate to http://localhost:8000 (or whatever host and port you have mapped) to test if the application is running properly.**
- **This is a way to ensure that your container is up and serving the application as expected.**

How to Deploy On Google Cloud Run

- **Create a google account**
- **goto gcp**
- **click on console to activate your GCP**
- **Tag your Docker image to upload it to Google Container Registry:**
- **docker tag docker.io/moodsphere/msdev gcr.io/[YOUR_PROJECT_ID]/moodsphere-backend**
- **Push the Docker image to Google Container Registry:**
- **docker push gcr.io/[YOUR_PROJECT_ID]/moodsphere-backend**

Deploy the image to Cloud Run:

- **gcloud run deploy --image gcr.io/[YOUR_PROJECT_ID]/moodsphere-backend --platform managed**
- **Follow the prompts to enable necessary APIs, set the region, and configure the service settings. Once deployed, Cloud Run will provide a URL to access your deployed backend.**
- **Congratulations! You have successfully deployed your backend application to Google Cloud Run.**

Contact Us and Feedback

We value your feedback and are here to assist you with any inquiries or concerns you may have regarding the MoodSphere application deployment process. Feel free to reach out to us via email at dynastydev23915@gmail.com. Whether you have questions about the deployment steps, encountered any issues during the process, or have suggestions for improvement, we welcome your input.

Additionally, if you have any feedback or ideas for enhancing the MoodSphere application, we would love to hear from you. Your insights play a crucial role in shaping the future development and functionality of our application. We are committed to delivering an exceptional user experience and greatly appreciate your input.

Thank you for choosing MoodSphere. We look forward to hearing from you!

Warm regards,
Team DevDynasty