

Installation Guide: Moodify

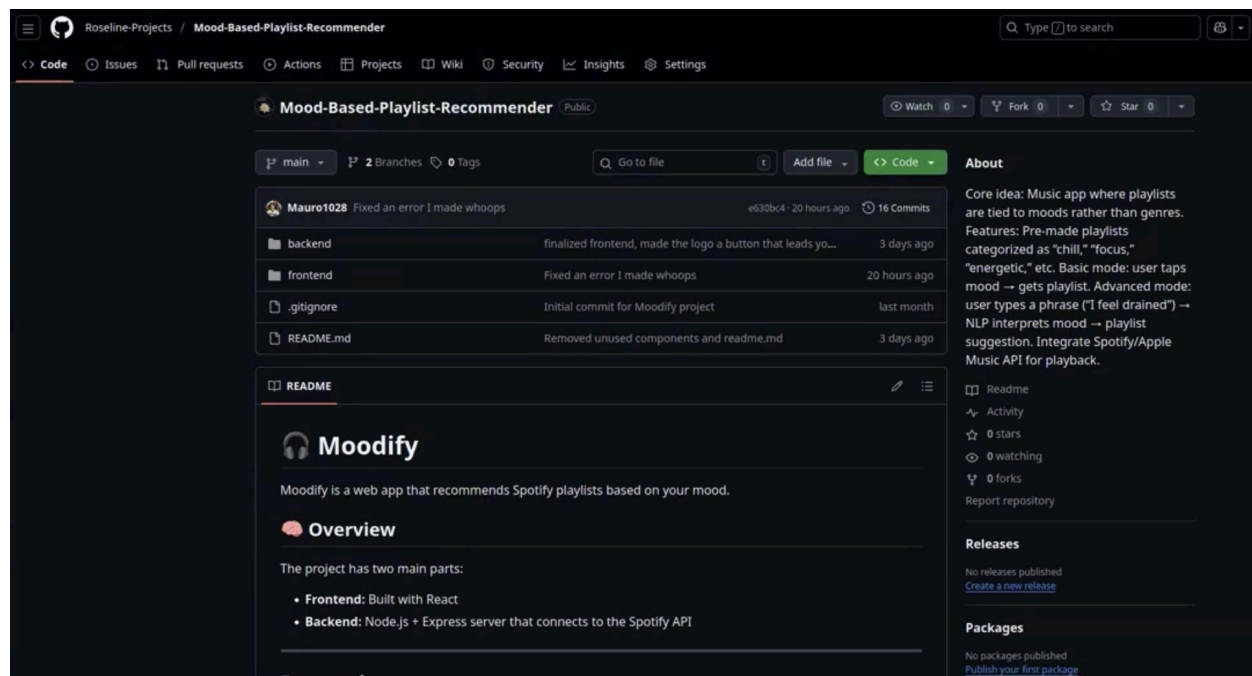
Introduction:

Moodify is a web-based application that helps university students quickly discover Spotify playlists that match their current mood. The system uses pattern-matching NLP to classify either a selected mood or a written mood description into one of five categories: Chill, Focus, Energetic, Happy, and Sad.

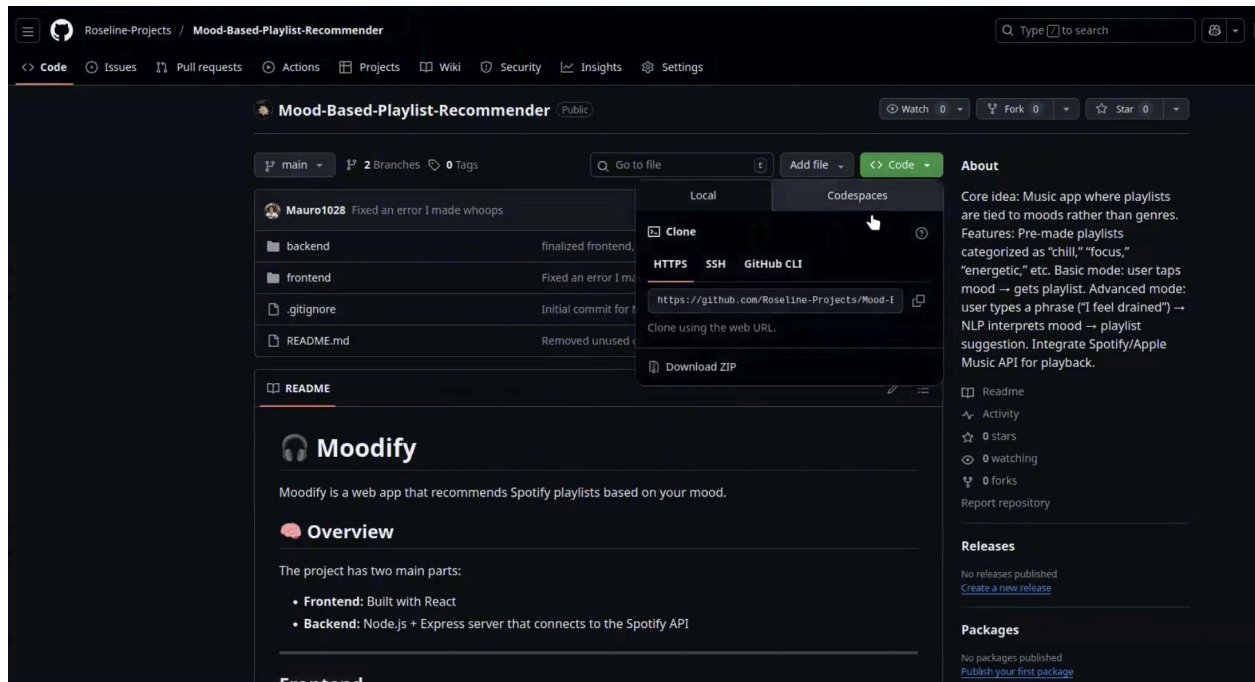
Based on this classification, Moodify retrieves up to 5 recommended playlists and 10 preview tracks per playlist from the Spotify Web API. This manual explains how to install Moodify on a new device.

Follow the steps with screenshots below:

1. Go into our GitHub page as shown below

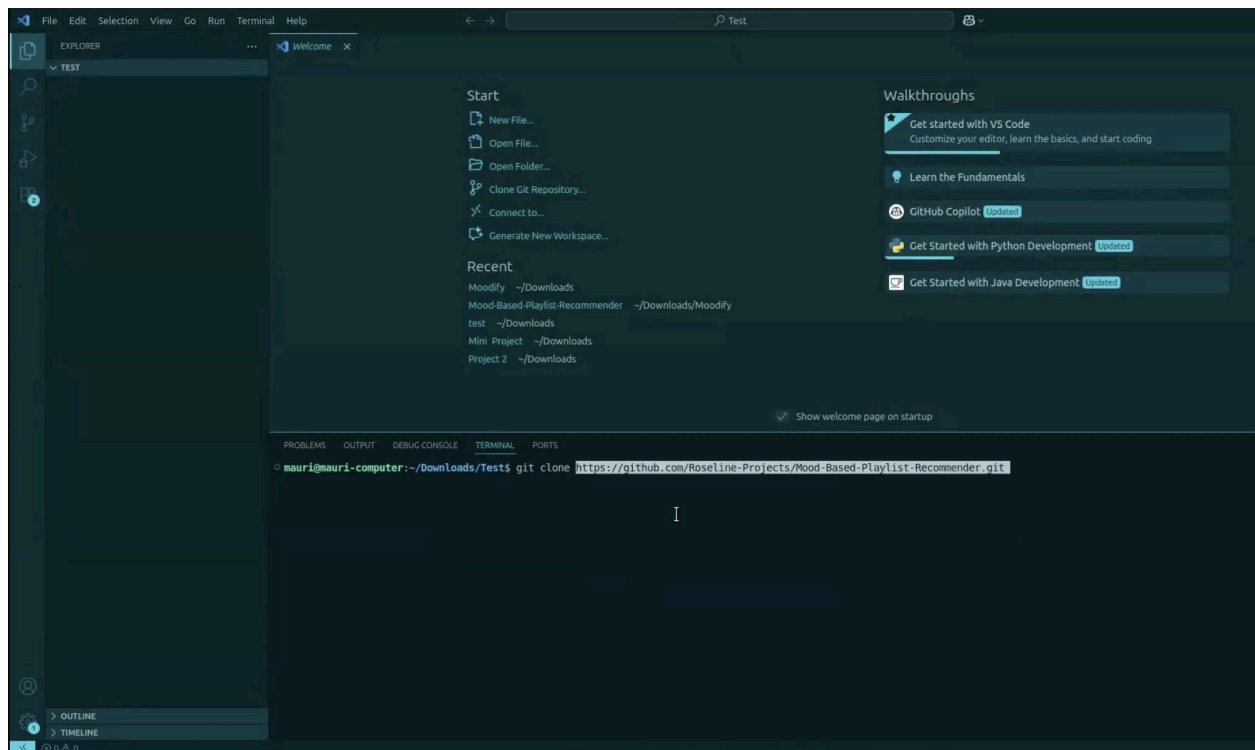


2. Copy GitHub url



3. Then go into your IDE of choice and set it in the location you would like the repository to be stored.

Below we use VSCode and typed the following command: `git clone github-url`



4. Once cloned you will then type `cd Mood-Based-Playlist-Recommender` as shown below

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
• mauri@mauri-computer:~/Downloads/Test$ git clone https://github.com/Roseline-Projects/Mood-Based-Playlist-Recommender.git
Cloning into 'Mood-Based-Playlist-Recommender'...
remote: Enumerating objects: 177, done.
remote: Counting objects: 100% (177/177), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 177 (delta 84), reused 129 (delta 47), pack-reused 0 (from 0)
Receiving objects: 100% (177/177), 485.30 KiB | 2.44 MiB/s, done.
Resolving deltas: 100% (84/84), done.
• mauri@mauri-computer:~/Downloads/Test$ cd Mood-Based-Playlist-Recommender
○ mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommender$
```

5. Then get the directory into frontend using `cd frontend` to install dependencies

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
• mauri@mauri-computer:~/Downloads/Test$ git clone https://github.com/Roseline-Projects/Mood-Based-Playlist-Recommender.git
Cloning into 'Mood-Based-Playlist-Recommender'...
remote: Enumerating objects: 177, done.
remote: Counting objects: 100% (177/177), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 177 (delta 84), reused 129 (delta 47), pack-reused 0 (from 0)
Receiving objects: 100% (177/177), 485.30 KiB | 2.44 MiB/s, done.
Resolving deltas: 100% (84/84), done.
• mauri@mauri-computer:~/Downloads/Test$ cd Mood-Based-Playlist-Recommender
• mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommender$ cd frontend
○ mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommender/frontend$
```

6. Install dependencies as shown below: `npm install`

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
mauri@mauri-computer:~/Downloads/Test$ git clone https://github.com/Roseline-Projects/Mood-Based-Playlist-Recommendder.git
Cloning into 'Mood-Based-Playlist-Recommendder'...
remote: Enumerating objects: 177, done.
remote: Counting objects: 100% (177/177), done.
remote: Compressing objects: 100% (116/116), done.
remote: Total 177 (delta 84), reused 129 (delta 47), pack-reused 0 (from 0)
Receiving objects: 100% (177/177), 485.30 KiB | 2.44 MiB/s, done.
Resolving deltas: 100% (84/84), done.
mauri@mauri-computer:~/Downloads/Test$ cd Mood-Based-Playlist-Recommendder
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommendder$ cd frontend
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommendder/frontend$ npm install
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'react-router@7.9.5',
npm WARN EBADENGINE   required: { node: '>=20.0.0' },
npm WARN EBADENGINE   current: { node: 'v18.19.1', npm: '9.2.0' }
npm WARN EBADENGINE }
npm WARN EBADENGINE Unsupported engine {
npm WARN EBADENGINE   package: 'react-router-dom@7.9.5',
npm WARN EBADENGINE   required: { node: '>=20.0.0' },
npm WARN EBADENGINE   current: { node: 'v18.19.1', npm: '9.2.0' }
npm WARN EBADENGINE }
(#####) : idealTree: timing idealTree Completed in 978ms
```

7. Once frontend installation is done, do the same in the backend using `cd backend` with the proper output shown below

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
run 'npm fund' for details
11 vulnerabilities (4 moderate, 7 high)
To address issues that do not require attention, run:
  npm audit fix
To address all issues (including breaking changes), run:
  npm audit fix --force
Run 'npm audit' for details.
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommendder/frontend$ cd ../
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommendder$ cd backend
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommendder/backend$ npm install
added 127 packages, and audited 128 packages in 777ms
24 packages are looking for funding
  run 'npm fund' for details
found 0 vulnerabilities
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommendder/backend$
```

8. Add a `.env` file in the backend folder in this format

```
$ .env.example x
Mood-Based-Playlist-Recommenderr > backend > $ .env.example
1 # .env.example
2 PORT=5001
3 MONGODB_URI=your_mongodb_connection_string_here
4 SPOTIFY_CLIENT_ID=your_spotify_client_id
5 SPOTIFY_CLIENT_SECRET=your_spotify_client_secret
6
```

9. After setting up the .env in the backend folder we will type `npm start` in the backend folder for the backend to start

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
added 127 packages, and audited 128 packages in 777ms
24 packages are looking for funding
  run `npm fund` for details
found 0 vulnerabilities
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommenderr/backend$
* History restored
● mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommenderr/backend$ cd ../
● mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommenderr$ cd backend
○ mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommenderr/backend$ npm start
> backend@1.0.0 start
> node server.js
[dotenv@17.2.3] injecting env (5) from .env -- tip: 🗝 add access controls to secrets: https://dotenvx.com/ops
[dotenv@17.2.3] injecting env (0) from .env -- tip: 🚫 prevent building .env in docker: https://dotenvx.com/prebuild
Server running on port 5001
MongoDB Connected: ac-qbkzbtj-shard-00-00.wkgkz8w.mongodb.net
I
```

It should run and have the output shown above

10. Then we create another terminal and set the directory into frontend and repeat the command `npm start` which should run the app

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
mauri@mauri-computer:~/Downloads/Test$ cd frontend
bash: cd: frontend: No such file or directory
mauri@mauri-computer:~/Downloads/Test$ cd Mood-Based-Playlist-Recommender
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommender$ cd frontend
mauri@mauri-computer:~/Downloads/Test/Mood-Based-Playlist-Recommender/frontend$ npm start
```

11. Once running, it should automatically open this page as shown below

