

COOK BOOK: YOUR VIRTUAL KITCHEN

ASSISTENT

Project submitted to the Bharathidasan University, Tiruchirappalli

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE

Submitted by

Name : NAVIN KUMAR.S

Reg . No: 24057011802111013

NM ID : F797A70A391A825BECA4DEB089434E0C

Under the Guidance of

Dr. M. RAJENDIRAN M.C.A., M.Phil., Ph. D



Department of Computer Science

Government Arts and Science College

Jayankondam-621801

November – 2025

Department of Computer Science

Government Arts and Science College

Jayankondam-621801



CERTIFICATE

This is to certify that the project entitled COOK BOOK : YOUR KITCHEN ASSISTANT

is a bonafide report done by NAVIN KUMAR.S Reg. No: 2405011802111013 in partial fulfillment of the requirements for the BACHELOR OF SCIENCE IN COMPUTER SCIENCE offered by the Department of Computer Science, Government Arts and Science College, Jayankondam during the academic year 2024 to 2027.

Faculty Mentor Head of the Department

The Viva-Voce held on …………… at the Department of Computer Science, Government Arts and Science College, Jayankondam.

Internal Examiner External Examiner

ACKNOWLEDGEMENT

I would like to thank Dr. M. RAJAMOORTHY, **M.Sc., M.Phil., Ph.D.,** Principal, Government Arts and Science College, Jayankondam for providing me an opportunity and environment that made the work possible.

I would like to express my deep sense of gratitude to Dr.G. KARTHIKEYAN, M.Sc., M.Phil., Ph.D., Head ,Department of Computer Science, Government Arts and Science College, Jayankondam for giving me valuable support to complete this study successfully.

I would like to extend my heartfelt thanks to my guide Dr. M.RAJENDIRAN, M.C.A.,M.Phil., Ph.D Department of Computer Science, Government Arts and Science College, Jayankondam for an efficient guidance throughout the course of my project work.

I also express my thanks to my parents, my friends and well-wisher for the encouragement and best wishes in the successful Completion of the project.

# COOK BOOK : YOUR KITCHEN ASSISTANT

## INTRODUCTION



**1**

Project Title: COOK BOOK : YOUR KITCHEN ASSISTANT

Team ID : NM2025TMID34957

Team Leader : [NAVINKUMAR.S&navinkumar.s377@gmail.com](mailto:NAVINKUMAR.S&navinkumar.s377@gmail.com)

Team Members:

1. [ABILASH.S&abi70027851@gmail.com](mailto:ABILASH.S&abi70027851@gmail.com)
2. [JAISUTHIN.M&suthinjaisuthin@gmai.com](mailto:JAISUTHIN.M&suthinjaisuthin@gmai.com)
3. [BALAJI.P&balaji790468@gmail.com](mailto:BALAJI.P&balaji790468@gmail.com)
4. [KARUPPAIYA.A&askaruppu9@gmail.com](mailto:KARUPPAIYA.A&askaruppu9@gmail.com)

# PROJECT OVERVIEW



**2**

Purpose: **COOK BOOK : YOUR VIRTUAL KITCHEN ASSISTANT** is a personalized music companion that helps users discover, enjoy, and even generate music matching their mood and activity. It combines intelligent music recommendations, real time streaming, and interactive features to create a unique, engaging listening experience.

## Key Features

* Personalized music recommendations based on listening habits, mood, and context.
* High-quality music streaming and playlist management with seamless cross-device playback.
* AI-powered “melodic companion”: suggests songs, creates custom playlists, and can generate original melodic snippets.
* Mood-based music generation: ambient background music or AI-generated tracks for study, workout, or relaxation.
* Social &community features: collaborative playlists , follow friends ,share favorite tracks.
* Secure user authentication & profiles with email/password or social logins.
* Admin control panel : manage content, users, and analytics.

# SYSTEM ARCHITECTURE



**3**

## Frontend

* Technology Stack: React.js (or Angular/Vue ) for building an interactive and responsive interface.
* Styling: Tail wind CSS/Boot strap for layout and UI design.

## Backend

* Technology Stack: Node.js with Express.js(or Python Flask/Dj ango ) for server logic and APIs.
* Authentication : JWT (JSON Web Tokens)for secure login and session handling.

## Database

* Preferred: Mongo DB (No SQL) for flexible, document- based storage.
* Alternative: My SQL/Post agree SQL relational structure quire

# SETUP INSTRUCTIONS



**4**

## Prerequisites

Software Requirements

* Node.js (back end server and package management)
* Express.js/Flask/Dango (depending on backend choice)
* Mongo DB/My SQL(data base management)
* React.js/Angular/Vue.js(front end development)
* Git for version control
* Visual Studio Code (VS Code) or any suitable IDE

System Requirements

* OS: Windows/Linux/mac OS
* RAM: Minimum 4GB (8GB recommended )
* Processor: Intel i3/AMD equivalent (i5 or higher recommended)
* Storage: Atleast 500 MB free space

Additional Dependencies

* Npm (Node Package Manager)
* Modern browser(Chrome/Firefox/Edge)
* Postman/Thunder Client for API testing

## Installation Steps

Clone the Repository

Git clone<repository\_ url> cd rhythmic tunes

Front end Setup

cd client npm install npm start

Back end Setup

cd../server npm install npm start

1. ​
2. Data base Setup
   * Install and start Mongo DB (or My SQL).
   * Create a database named rhythmic tunes\_ db.
   * Configure connection detail sin server/config/ db.js orina. envfile.
3. Access the Application
   * Open: [http://localhost:3000](http://localhost:3000/)
   * Register or login to explore features.

# FOLDER STRUCTURE



**5**

│

├──client/ #Frontend

│├──public/ #Static files (HTML, images ,icons)

│├──src/

││├──components/ #Reusable UI components

││├──pages/ #Pages(Landing, Dashboard, Track Details)

││├──assets/ #Images, styles, fonts

││├──services/ #API call sto backend

││└──App.js #Root React component

│└──package. json

│

├──server/ #Backend

│├──routes/ #API route definitions

│├──models/ #Data base models

│├──controllers/ #Route logic

│├──config/ #DB connection &environment variables

│└──server.js #Enter point

│

├──database/ #Data base scripts/se\add ata

├──docs/ #Project documentation

├──.env #Environment variables

├──README.md #Overview & instructions

└──package. json

# RUNNING THE APPLICATION



**6**

Frontend

Cd client npm start

Backend

Cd server npm start

Access via:[http://localhost:3000](http://localhost:3000/)

# API DOCUMENTATION



**7**

## User APIs

* POST/api / user/register–Register an ew user
* POST/api/user/login–Auth ent cate user& return JWT token
* GET/api/user/profile/:id–Fetch user profile
* PUT/api/user/update/:id–Update user details
* DELETE/api/user/:id–Delete user account

## Project APIs

* POST/api/projects/create–Create an project/track
* GET/api/projects/:id–Fetch project/track details
* GET/api/projects–Fet hall projects/tracks
* PUT/api/projects/:id–Update project/track info
* DELETE/api/projects/:id–Remove project/track

## Chat APIs

* POST/api/chat/send–Send a message
* GET/api/chat/:user Id–Fetch chat history
* GET/api/chat/conversations/:user Id–List all conversations
* DELETE/api/chat/:chat Id–Delete a chat message

# AUTHENTICATION



**8**

## Flow

* Users (listeners ,artists, a dmins) login with email or username.
* Server sues JWT stored (HTTP- only cookie or local storage).
* JWT contains user role information for Role-Based Access Control (RBAC).

## Registration & Login

* Password has using bord pt.
* After login, signed JWT returned to client.

## Middle ware Protection

* Custom middle ware verifies JWT on protected end points:
  + Uploading music (for artists)
  + Managing playlists
  + Access singed min dashboard

Invalid/ missing to kens result in an error.

## Role-Based Access

* Listeners: Stream music, create playlists, follow artists.
* Artists: Upload managment racks, view analytics.
* Admins : Manage users, moderate content.

## Token Refresh &Security

* Access tokens short- lived (15–30 min).
* Refresh tokens used for silent-authentication.
* Tokens signed with strong secret keys.

## Optional Enhancements

* Social logins (Google, Apple) via O Auth2.
* Two-Factor Authentication (2FA) for artists and admins.

# USER INTERFACE



**9**

## Landing Page

* + Hero section with tag line:*“Your Melodic Companion–Discover, Listen & Create”*
  + Trending songs & artists
  + Preview player (30-secpreview)
  + Call-to-action: Sign Up, Start Listening, Upload Track

## Listener Dashboard

. Personalized recommendations

* + Playlist creation & sharing
  + Followed-artist updates
  + Mini “Now Playing” widget
  + Favorites & Recently Played

## Artist Dashboard

* + Upload & manage tracks
  + View analytics (plays ,likes ,followers)
  + Monetization tools(exclusive tracks/subscriptions)
  + Fan engagement & profile customization

## Admin Panel

* + User management & content moderation
  + Platform analytics& revenue tracking
  + Subscription management & system settings

## Track Details Page

* + High-quality audio streaming with wave form visualization
  + Synchronized lyrics section
  + Artist info, comments & likes
  + Related track suggestions

## Optional Mobile App Features

* + Offline mode for downloaded songs
  + Push notifications for new releases

10TESTING

## Manual Testing During Milestones

* + Feature checks: after completing modules (streaming engine, play list management, artist dashboard).
  + User journey validation: Listener (signup→ play→ add to playlist), Artist (upload track →view analytics), Admin (moderate content).
  + Cross-browser/device testing on Chrome, Firefox, Edge, Safari.
  + Negative testing : invalid inputs & error handling.
  + Security checks: ensure JWT- protected routes are in accessible without valid tokens

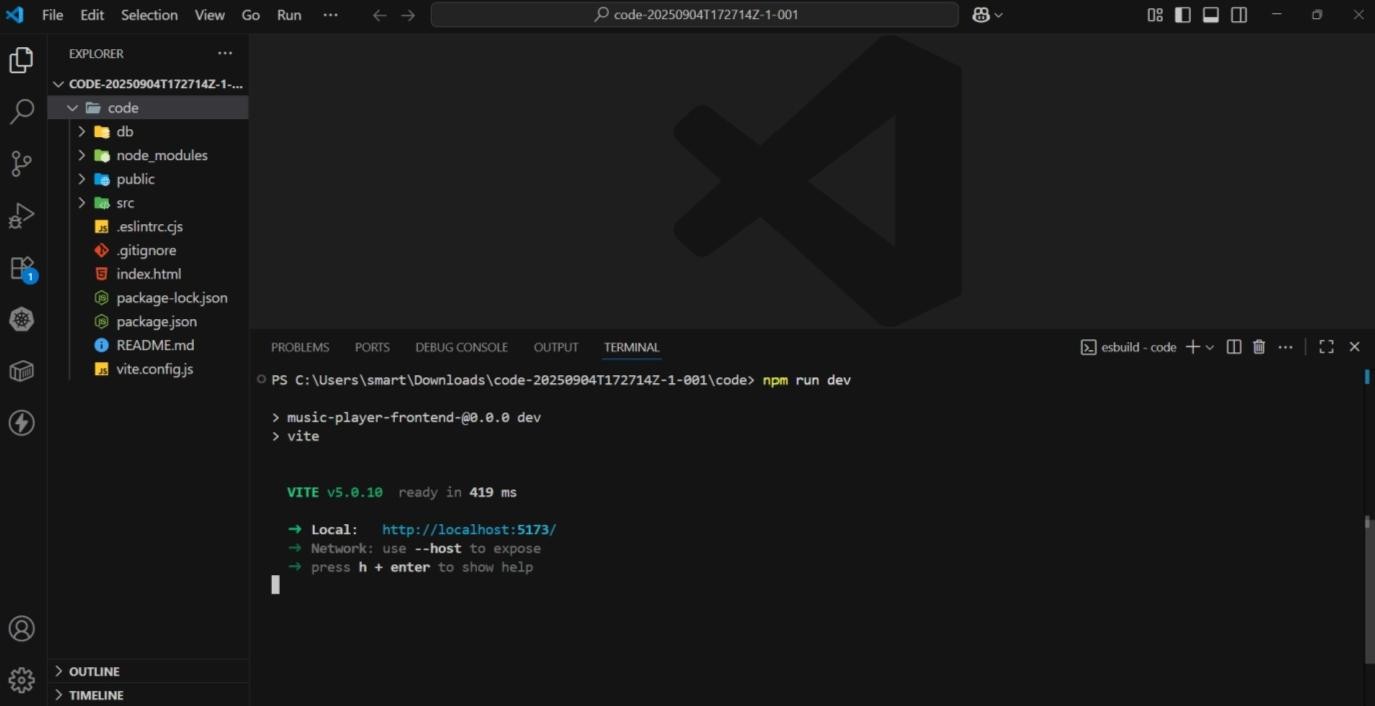
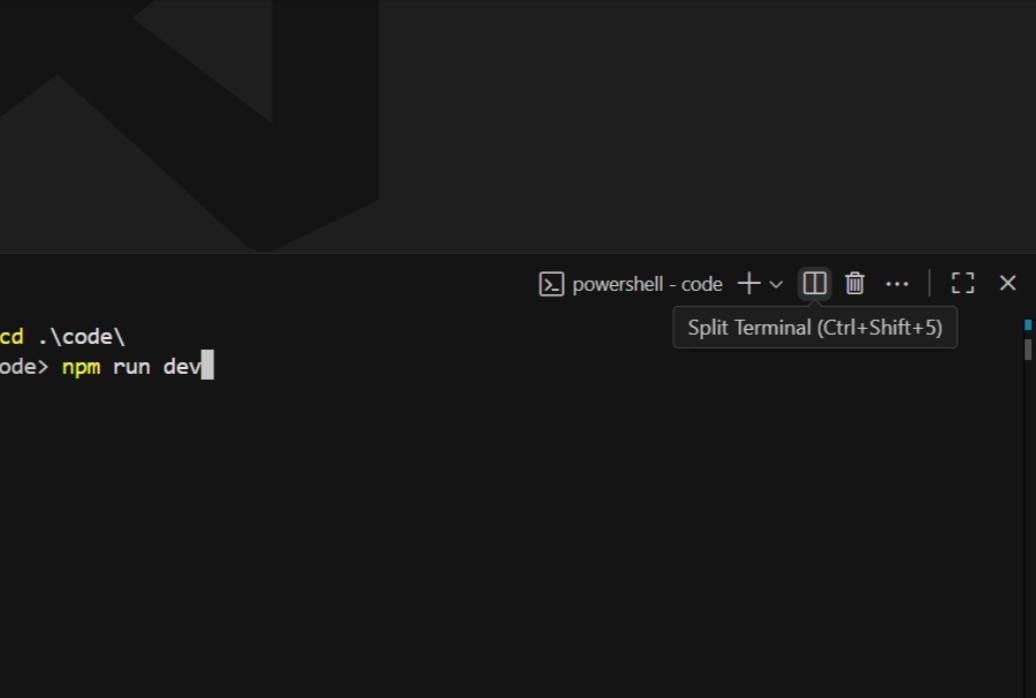
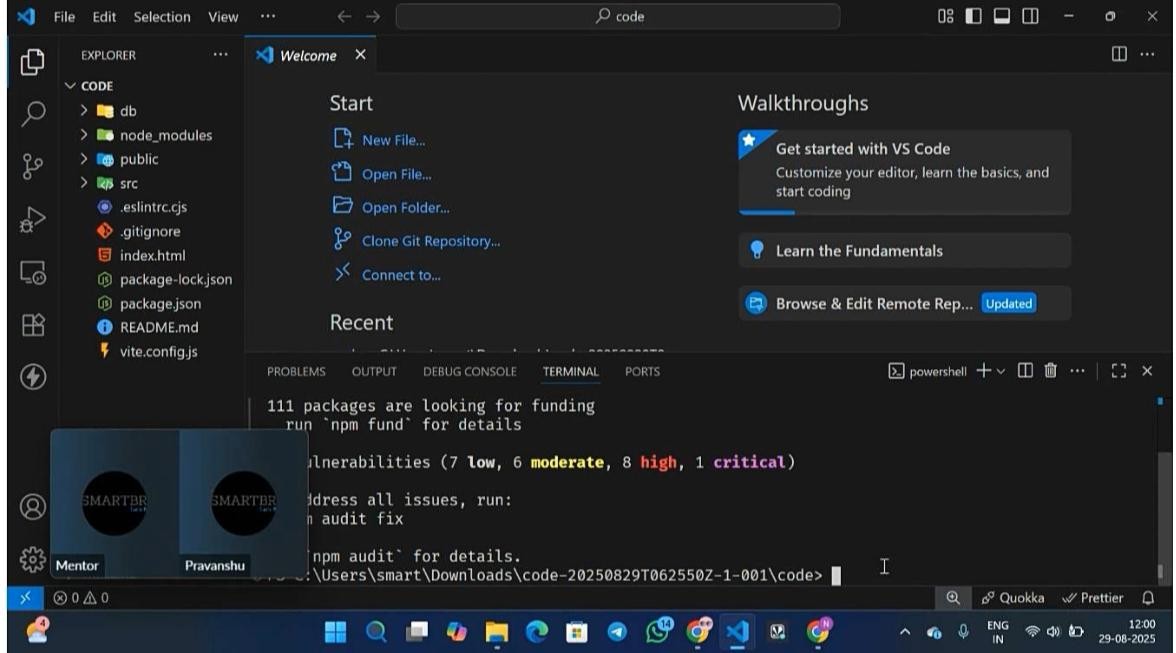
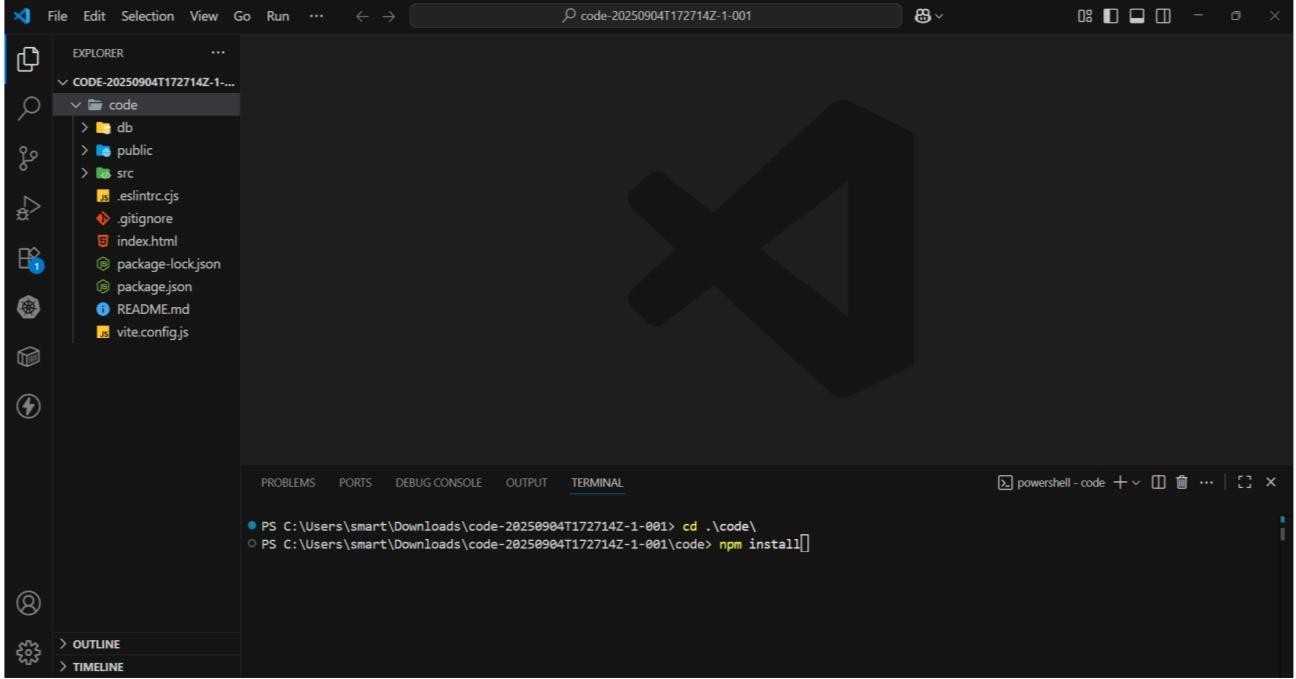
## Tools

* + Postman: Test RESTAPI s for JWT, uploads, playlists.
  + Chrome Dev Tools: Debug frontend, monitor streaming performance, run Light house audits.

## Outcome

* + Early bug detection ensures smooth playback, secure login, and reliable music upload/streaming.

11. SCREENSHOTS OR DEMO



12. KNOWN ISSUES

## Audio & Streaming

* + Occasional buffering on slow networks.
  + Wave form visualizations one times lags behind play back.

## UI/UX

* + Mobile responsiveness gaps (e.g., Artist Dashboard charts).
  + Dark mode contrast issues.

## Authentication & Security

* + Token refresh occasionally forces full re-login.
  + Google/Apple login the ration pending.

## Artist Dashboard

* + Bulk upload limit (>5tracks) may fail with large files.
  + Analytics refresh every 10–15 minutes, not live.

## Admin Panel

* + Reported track scan take up to appearing moderation.

## General Performance

* + First-time load is slow on low-end devices.
  + Brow searching one times shows out dated playlist art work.

Planned Fixes

* + Optimize CDN and caching strategy.
  + Improve mobile responsiveness and dark mode.
  + Enhance real-time analytics.
  + Complete O Auth (Google/Apple) integration.

# 13 FUTURE ENHANCEMENTS

## Advanced Music Discovery

* + AI-driven recommendations based on mood and listening habits.
  + Smart play lists auto-generated and updated daily.

## Enhanced Artist Tools

* + Monetization options: sell exclusive tracks or offer sub scriptiontiers.
  + Collaboration platform for co-creating music.

## Social & Community Features

* + Live listening parties.
  + Song-level comments and genre-based forums.

## Mobile & Offline

* + Dedicated iOS/Android apps with push notifications.
  + Offline mode for downloaded playlists.

## 

## Advanced Streaming & Audio Quality

* + Lossless/Hi-Res audio for premium users.
  + Adaptive bit rate streaming for seamless playback.

## Security & Authentication

* + Two-Factor Authentication (2FA).
  + O Auth login via Google, Apple, or Spotify.

## Analytics & Insights

* + Real-time listener analytics and recommendation feedback.

## Integrations & Partnerships

* + Smart speaker support (Alexa,GoogleHome).
  + Event & ticketing integration for concerts and virtual shows