KukuFm AudioBook Review and Rating System

Sudip Kumar (B20CS072)

kumar.234@iitj.ac.in

Table of Contents

1. Overview

This web application allows users to browse and search for audiobooks, listen to them, and filter on audiobooks by genre, author, and rating. Users can also log in and sign up.
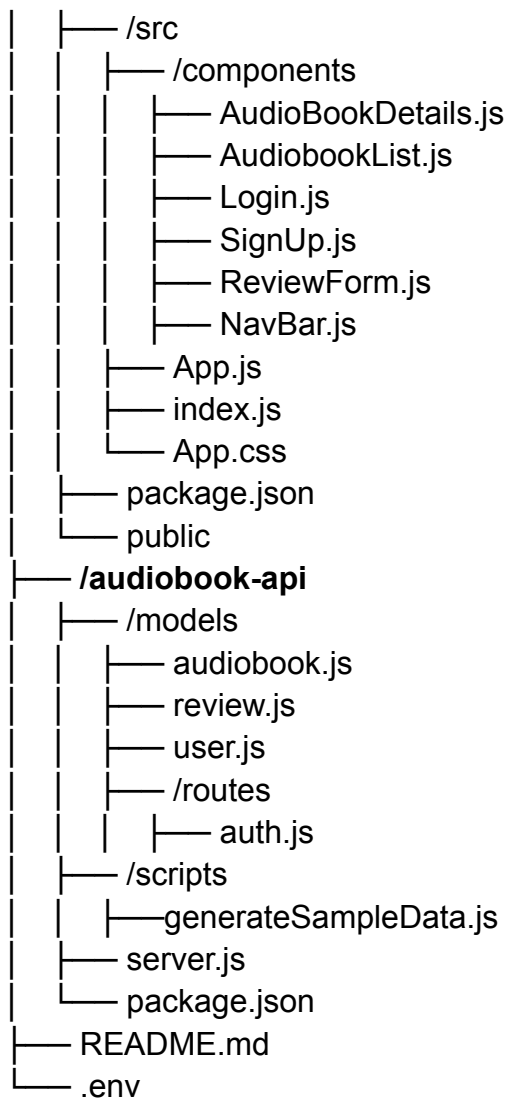
2. Application Architecture

Frontend

- React: The core library for building the user interface.
- React Router: For client-side routing.
- Material-UI: For UI components and responsive design.
- Axios: For making HTTP requests to the backend API.
- React Stars: For displaying star ratings.
- Custom Components: Components like NavBar, AudiobookList, and others.

Backend
- Node.js: JavaScript runtime for building the server-side application.
- Express.js: Web framework for Node.js to create the API endpoints.
- MongoDB: NoSQL database to store audiobook and user data.
- Mongoose: ORM for MongoDB to interact with the database.
- JWT: For authentication and authorization.

Directory structure:

├──**/audiobook-app**

```
│   ├── /src
│   │   ├── /components
│   │   │   ├── AudioBookDetails.js
│   │   │   ├── AudiobookList.js
│   │   │   ├── Login.js
│   │   │   ├── SignUp.js
│   │   │   ├── ReviewForm.js
│   │   │   ├── NavBar.js
│   │   ├── App.js
│   │   ├── index.js
│   │   └── App.css
│   ├── package.json
│   └── public
├── /audiobook-api
│   ├── /models
│   │   ├── audiobook.js
│   │   ├── review.js
│   │   ├── user.js
│   │   ├── /routes
│   │   │   ├── auth.js
│   ├── /scripts
│   │   ├──generateSampleData.js
│   ├── server.js
│   └── package.json
├── README.md
└── .env
```

3. API Usage

Base URL**—>**   **http://localhost:5000**

Endpoints
   1. **Audiobooks**
      GET /audiobooks
            ● Fetch all audiobooks.
            ● Example Response:
            ○ _id of the book
            ○ Title
            ○ Author

- ○ Genre
  - ○ Rating
  - ○ coverImage
  - ○ Audio

```
[
  {
    "_id": "60c72b2f9b1e8a5f1f1b6a3d",
    "title": "The Great Gatsby",
    "author": "F. Scott Fitzgerald",
    "genre": "Fiction",
    "rating": 4.5,
    "coverImage": "url-to-image",
    "audio": "url-to-audio"
  }
  ...
]
```

2. **User**
POST /signup
- ● Register a new user.
- ● Request Body
  - ○ Name
  - ○ Email
  - ○ Password
  - ○ Avatar image
- ● Response body
  - ○ Message
  - ○ User{ Id, name, email, image}

POST /login
- ● Log in an existing user.
- ● Request Body
  - ○ Email
  - ○ Password
- ● Response body
  - ○ Message
  - ○ Token

○ User{ Id, name, email, image}

Deployment Steps:

Prerequisites
- Node.js and npm installed.
- MongoDB installed or a cloud MongoDB instance.

Backend Deployment

1. Clone the Repository
2. Install Dependencies
3. Configure Environment Variables
   a. Create a .env file in the /backend directory and add the following:
   b. MONGO_DB_URL =.<your database url>
   c. PORT=5000
4. Start the Server-> npm run start

Frontend Deployment
1. Navigate to the Frontend Directory ->cd ../audiobook-app
2. Install Dependencies
3. Start the Frontend -> npm run start

Url of the live website:

[React App (kuku-fm-rating-and-review-web-api.vercel.app)](#)

Github link to the project

https://github.com/ksudip/KukuFm-Rating-and-Review-web-api.git