**Media Verse – Full Stack Application**

**Overview**

Media Verse is a full-stack web application that allows users to add, view, edit, and delete their favorite media content — including movies, series, or other entertainment forms.  
It features user authentication (Signup/Login), secure media management, and a clean modern UI built using React, TypeScript, Express, Prisma, and MySQL.

**Features**

* User Authentication (Signup/Login)
* Add Media with title, director, budget, duration, location, and year
* Edit and Delete Media Entries
* Upload Images for each media item
* Separate Backend and Frontend
* Data stored securely in MySQL
* API integration with Axios
* Form Validation and Alerts

**Tech Stack**

**Frontend**

* React (TypeScript)
* Vite
* Axios
* Framer Motion
* Tailwind CSS

**Backend**

* Node.js (Express)
* Prisma ORM
* MySQL Database
* Multer (for image uploads)
* JWT Authentication
* Zod (for validation)

**Project Structure**

FavoriteMediaApp/

│

├── backend/

│ ├── src/

│ │ ├── controllers/

│ │ ├── routes/

│ │ ├── middleware/

│ │ ├── utils/

│ │ └── validators/

│ ├── prisma/

│ ├── uploads/

│ ├── package.json

│ └── tsconfig.json

│

├── frontend/

│ ├── src/

│ │ ├── components/

│ │ ├── pages/

│ │ ├── services/

│ │ └── types/

│ ├── public/

│ ├── package.json

│ └── tsconfig.json

│

└── README.md

**Setup Instructions**

**1. Clone the Repository**

git clone https://github.com/manognya66/FavoriteMediaApp.git

cd FavoriteMediaApp

**2. Setup the Backend**

cd backend

npm install

Configure Environment Variables

Create a .env file inside /backend:

DATABASE\_URL="mysql://user:password@localhost:3306/media\_verse"

JWT\_SECRET="your\_jwt\_secret"

PORT=5000

Run Database Migrations

npx prisma migrate dev --name init

Start the Backend Server

npm run dev

**3. Setup the Frontend**

cd ../frontend

npm install

npm run dev

Frontend runs by default on http://localhost:5173  
Backend runs on http://localhost:5000

**User Authentication (Local Only)**

**The app isn’t hosted publicly — it runs locally.  
However, login and signup are required to use the application.**

**Example Credentials**

**Signup:**

name: test

email: test@gmail.com

password: test@123

**Login:**

email: test@gmail.com

password: test@123

**API Endpoints**

**Auth Routes**

| Method | Endpoint | Description |
| --- | --- | --- |
| POST | /api/auth/signup | Register a new user |
| POST | /api/auth/login | Login existing user |

**Media Routes**

| **Method** | **Endpoint** | **Description** |
| --- | --- | --- |
| GET | /api/media | Get all media for logged-in user |
| POST | /api/media | Add a new media item |
| PUT | /api/media/:id | Edit media item |
| DELETE | /api/media/:id | Delete media item |

**Example Media Entry**

{

"title": "Inception",

"type": "Movie",

"director": "Christopher Nolan",

"budget": "160 million USD",

"duration": "148 minutes",

"year": "2010",

"location": "Los Angeles"

}

**Notes**

* Make sure MySQL is running locally.
* The backend and frontend must both be active for the app to work.
* Uploaded images are stored in /backend/uploads.
* Only authenticated users can access media routes.

**Future Improvements**

* Role-based access (Admin/User)
* Cloud deployment (Vercel + Render)
* Enhanced UI with filtering and search
* Dark/Light mode support

**Author**

R. Shashank Manognya  
Frontend Developer  
GitHub: [manognya66](https://github.com/manognya66)