

```

Vidify/
├─ pages/
│   ├─ _app.tsx
│   ├─ index.tsx
│   └─ api/
│       ├─ videos.ts
│       ├─ likes.ts
│       └─ comments.ts
├─ components/
│   ├─ VideoCard.tsx
│   └─ UploadModal.tsx
├─ lib/
│   └─ mockDB.ts
├─ prisma/
│   └─ schema.prisma
├─ public/
│   └─ placeholder.mp4
├─ styles/
│   └─ globals.css
├─ .gitignore
├─ .env.example
├─ package.json
├─ tsconfig.json
├─ next.config.js
└─ tailwind.config.js

```

---

## File Contents

### pages/\_app.tsx

```

import '../styles/globals.css' import type { AppProps } from 'next/app' export default
function App({ Component, pageProps }: AppProps) { return <Component {...pageProps} /> }

```

### pages/index.tsx

```

import { useState, useEffect } from 'react' import VideoCard from
'../components/VideoCard' import UploadModal from '../components/UploadModal' import {
getVideos } from './api/videos' export default function Home() { const [videos, setVideos]
= useState<any[]>([]) const [showUpload, setShowUpload] = useState(false) useEffect(() =>

```

```
{ setVideos(getVideos()) }, []) const handleUpload = (video: any) => { setVideos([video,
...videos]) } return ( <div className="min-h-screen bg-gray-900 text-white flex flex-col
items-center"> <button className="mt-4 px-4 py-2 bg-blue-600 rounded" onClick={() =>
setShowUpload(true)} > Upload Video </button> {showUpload && <UploadModal onClose={() =>
setShowUpload(false)} onUpload={handleUpload} />} <div className="mt-6 w-full max-w-md">
{videos.map((v) => ( <VideoCard key={v.id} video={v} /> ))} </div> </div> ) }
```

## pages/api/videos.ts

```
import { getVideos, addVideo } from '../lib/mockDB' export default function
handler(req: any, res: any) { if (req.method === 'GET') {
res.status(200).json(getVideos()) } else if (req.method === 'POST') { const video =
req.body addVideo(video) res.status(201).json(video) } else { res.status(405).end() } }
```

## pages/api/likes.ts

```
export default function handler(req: any, res: any) { res.status(200).json({ message:
'Likes API (mocked)' }) }
```

## pages/api/comments.ts

```
export default function handler(req: any, res: any) { res.status(200).json({ message:
'Comments API (mocked)' }) }
```

## components/VideoCard.tsx

```
import { useState } from 'react' export default function VideoCard({ video }: any) {
const [likes, setLikes] = useState(video.likes || 0) return ( <div className="mb-6 border
rounded overflow-hidden bg-gray-800"> <video src={video.url} controls className="w-full"
/> <div className="p-2 flex justify-between items-center"> <span>{video.title}</span>
<button onClick={() => setLikes(likes + 1)} className="text-red-500"> ❤️ {likes}
</button> </div> </div> ) }
```

## components/UploadModal.tsx

```
import { useState } from 'react' export default function UploadModal({ onClose, onUpload
}: any) { const [file, setFile] = useState<File | null>(null) const [title, setTitle] =
useState('') const handleSubmit = () => { if (!file) return const newVideo = { id:
Date.now(), title, url: URL.createObjectURL(file), likes: 0, } onUpload(newVideo)
onClose() } return ( <div className="fixed inset-0 bg-black bg-opacity-70 flex items-
center justify-center z-50"> <div className="bg-gray-900 p-6 rounded w-96"> <h2
className="text-lg mb-4">Upload Video</h2> <input type="file" onChange={(e) =>
setFile(e.target.files?.[0] || null)} className="mb-2 w-full" /> <input type="text"
placeholder="Title" value={title} onChange={(e) => setTitle(e.target.value)}
className="mb-2 w-full p-1 rounded text-black" /> <div className="flex justify-end gap-2
```

```
mt-4"> <button onClick={onClose} className="px-3 py-1 bg-gray-700 rounded">Cancel</button>
<button onClick={handleSubmit} className="px-3 py-1 bg-blue-600 rounded">Upload</button>
</div> </div> </div> ) }
```

## lib/mockDB.ts

```
let videos = [ { id: 1, title: 'Sample Video', url: '/placeholder.mp4', likes: 0 }, ]
export function getVideos() { return videos } export function addVideo(video: any) {
  videos = [video, ...videos] }
```

## prisma/schema.prisma

```
datasource db { provider = "postgresql" url = env("DATABASE_URL") } generator client {
  provider = "prisma-client-js" } model User { id String @id @default(cuid()) email String
  @unique name String? } model Video { id String @id @default(cuid()) title String url
  String likes Int @default(0) uploader User? @relation(fields: [uploaderId], references:
  [id]) uploaderId String? }
```

## styles/globals.css

```
@tailwind base; @tailwind components; @tailwind utilities; body { margin: 0; font-family:
  sans-serif; }
```

## .gitignore

```
node_modules/ .next/ out/ dist/ npm-debug.log* .env uploads/ .DS_Store .vscode/
```

## .env.example

```
DATABASE_URL=postgresql://user:password@host:5432/vidify
JWT_SECRET=replace_with_a_strong_secret STORAGE_PROVIDER=local STORAGE_PATH=uploads/
```

## package.json

```
{ "name": "vidify", "version": "1.0.0", "scripts": { "dev": "next dev", "build": "next
  build", "start": "next start" }, "dependencies": { "next": "13.5.2", "react": "18.2.0",
  "react-dom": "18.2.0" }, "devDependencies": { "tailwindcss": "^3.3.3", "typescript":
  "^5.2.2" } }
```

After extracting, open a terminal in `vidify/` and run:

```
npm install npm run dev
```

Then visit:

```
http://localhost:3000
```