// firebase.ts

Import { initializeApp } from “firebase/app”;

Import { getStorage } from “firebase/storage”;

Const firebaseConfig = {

apiKey: process.env.NEXT\_PUBLIC\_FIREBASE\_API\_KEY,

authDomain: process.env.NEXT\_PUBLIC\_FIREBASE\_AUTH\_DOMAIN,

projectId: process.env.NEXT\_PUBLIC\_FIREBASE\_PROJECT\_ID,

storageBucket: process.env.NEXT\_PUBLIC\_FIREBASE\_STORAGE\_BUCKET,

messagingSenderId: process.env.NEXT\_PUBLIC\_FIREBASE\_MESSAGING\_SENDER\_ID,

appId: process.env.NEXT\_PUBLIC\_FIREBASE\_APP\_ID,

};

Const app = initializeApp(firebaseConfig);

Const storage = getStorage(app);

Export { storage };

NEXT\_PUBLIC\_FIREBASE\_API\_KEY=your\_api\_key

NEXT\_PUBLIC\_FIREBASE\_AUTH\_DOMAIN=your\_project.firebaseapp.com

NEXT\_PUBLIC\_FIREBASE\_PROJECT\_ID=your\_project\_id

NEXT\_PUBLIC\_FIREBASE\_STORAGE\_BUCKET=your\_project.appspot.com

NEXT\_PUBLIC\_FIREBASE\_MESSAGING\_SENDER\_ID=your\_messaging\_sender\_id

NEXT\_PUBLIC\_FIREBASE\_APP\_ID=your\_app\_id

// pages/api/upload-image.ts

Import type { NextApiRequest, NextApiResponse } from ‘next’;

Import { getStorage, ref, uploadString } from ‘firebase/storage’;

Import { initializeApp } from ‘firebase/app’;

Const firebaseConfig = {

apiKey: process.env.NEXT\_PUBLIC\_FIREBASE\_API\_KEY,

authDomain: process.env.NEXT\_PUBLIC\_FIREBASE\_AUTH\_DOMAIN,

projectId: process.env.NEXT\_PUBLIC\_FIREBASE\_PROJECT\_ID,

storageBucket: process.env.NEXT\_PUBLIC\_FIREBASE\_STORAGE\_BUCKET,

messagingSenderId: process.env.NEXT\_PUBLIC\_FIREBASE\_MESSAGING\_SENDER\_ID,

appId: process.env.NEXT\_PUBLIC\_FIREBASE\_APP\_ID,

};

Const app = initializeApp(firebaseConfig);

Const storage = getStorage(app);

Export default async function handler(req: NextApiRequest, res: NextApiResponse) {

If (req.method === ‘POST’) {

Const { image, filename } = req.body;

Try {

Const storageRef = ref(storage, `uploads/${filename}`);

Await uploadString(storageRef, image, ‘data\_url’);

Res.status(200).json({ message: ‘Upload successful’ });

} catch (error) {

Res.status(500).json({ error: ‘Upload failed’ });

}

} else {

Res.status(405).end(); // Method Not Allowed

}

}

Import { useState } from “react”;

Export default function Upload() {

Const [image, setImage] = useState<File | null>(null);

Const [status, setStatus] = useState(“”);

Const handleUpload = async () => {

If (!image) return;

Const reader = new FileReader();

Reader.onloadend = async () => {

Const base64 = reader.result;

Const res = await fetch(“/api/upload-image”, {

Method: “POST”,

Headers: { “Content-Type”: “application/json” },

Body: JSON.stringify({

Image: base64,

Filename: image.name,

}),

});

Const data = await res.json();

setStatus(data.message || data.error);

};

Reader.readAsDataURL(image);

};

Return (

<div className=”p-8”>

<h1 className=”text-xl font-bold mb-4”>Upload Image to Firebase</h1>

<input type=”file” accept=”image/\*” onChange={€ => setImage(e.target.files?.[0] || null)} />

<button

onClick={handleUpload}

className=”mt-4 px-4 py-2 bg-blue-600 text-white rounded”

>

Upload

</button>

{status && <p className=”mt-4”>{status}</p>}

</div>

);

}

Import Link from “next/link”;

Export default function Home() {

Return (

<div className=”p-8”>

<h1 className=”text-2xl font-bold mb-4”>Smart Farming AI</h1>

<Link href=”/upload”>

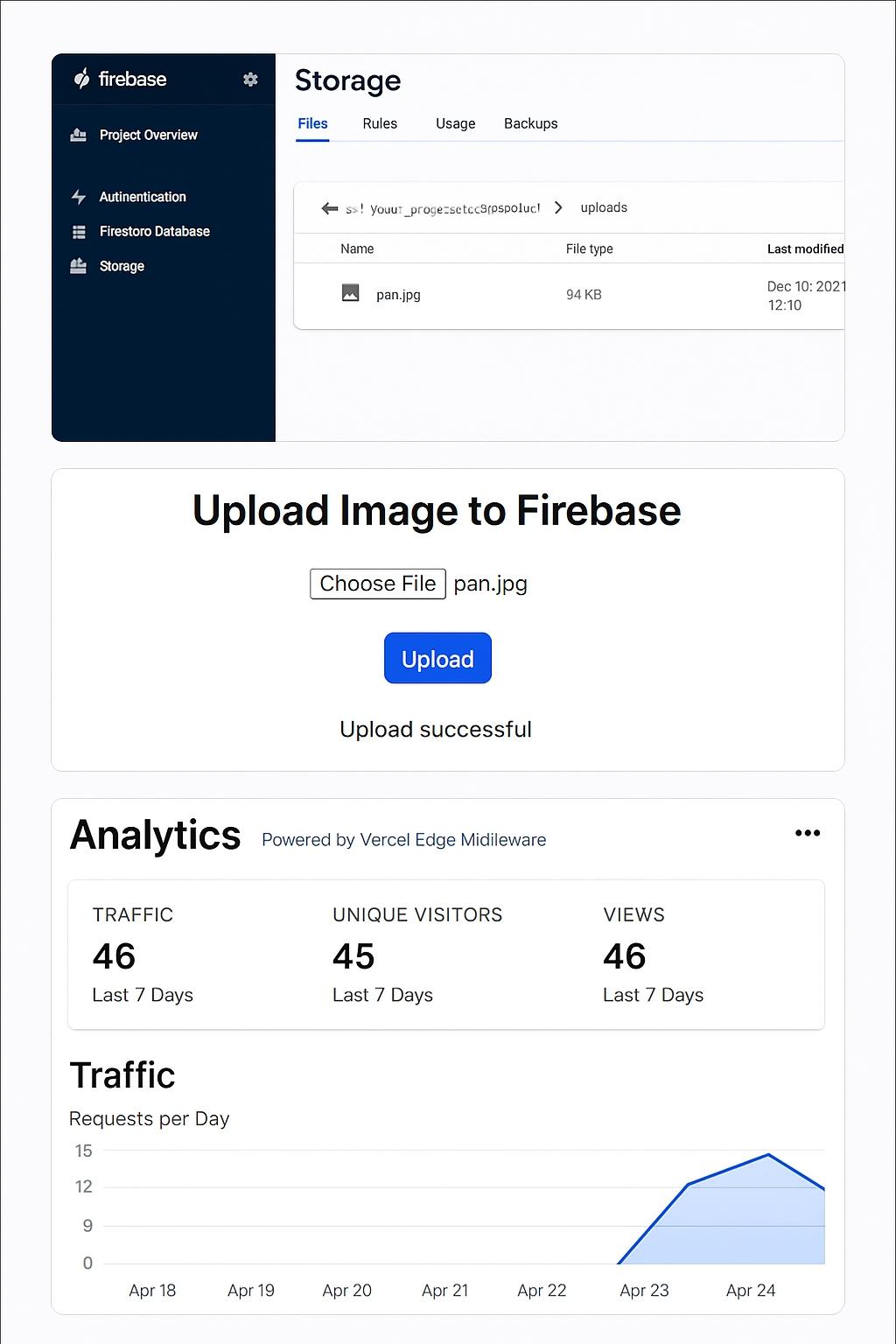
<a className=”text-blue-600 underline”>Go to Upload Page</a>

</Link>

</div>

);

}

Github:<https://github.com/ShreenithiE/MVP-internship>

Versel link:<https://mvp-internship-theta.vercel.app/>