Ex no:6 **Implementation of Storage as a service using Dropbox APIs**

Date:

Aim:

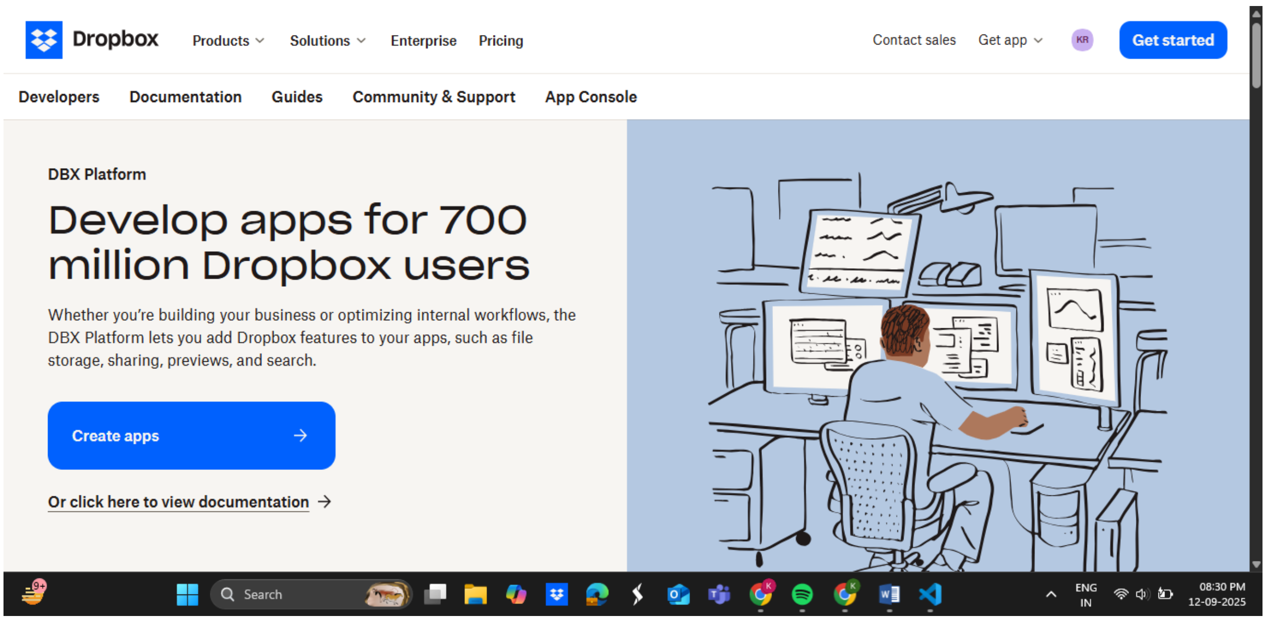
To integrate Storage as a service into application using Dropbox APIs.

Code:

Server.js:

```
require("dotenv").config();
const express = require("express");
const mongoose = require("mongoose");
const cors = require("cors");
const bcrypt = require("bcryptjs");
const multer = require("multer");
const fs = require("fs");
const { Dropbox } = require("dropbox");
const app = express();
app.use(cors());
app.use(express.json());
mongoose.connect("mongodb://127.0.0.1:27017/prepare", {
  useNewUrlParser: true,
  useUnifiedTopology: true,
})
.then(() => console.log(“ MongoDB Connected to 'prepare' database"))
.catch(err => console.error("MongoDB Error:", err));
const dbx = new Dropbox({ accessToken: process.env.DROPBOX_ACCESS_TOKEN });
const upload = multer({ dest: "uploads/" });
const UserSchema = new mongoose.Schema({
  name: String,
  email: { type: String, unique: true },
  password: String});
const User = mongoose.model("User", UserSchema);
const FeedbackSchema = new mongoose.Schema({
  text: { type: String, required: true },
  fileName: { type: String },
  dropboxPath: { type: String },
  createdAt: { type: Date, default: Date.now }});
const Feedback = mongoose.model("Feedback", FeedbackSchema);
app.get("/", (req, res) => {
  res.send("Welcome to Prepare.com API ");
});
app.post("/feedback", upload.single("file"), async (req, res) => {
  try {
    const { text } = req.body;
    if (!text) return res.json({ success: false, message: "Feedback cannot be empty" });
    let fileName = null;
    let dropboxPath = null;
    if (req.file) {
      const fileContent = fs.readFileSync(req.file.path);
      fileName = req.file.originalname;
      dropboxPath = "/feedback_uploads/" + fileName;
      await dbx.filesUpload({
        path: dropboxPath,
        contents: fileContent,
        mode: "overwrite"
      });
      fs.unlinkSync(req.file.path);  }
    const feedback = new Feedback({ text, fileName, dropboxPath });
    await feedback.save();
    res.json({ success: true, message: "Thank you for your feedback!", feedback });
  } catch (err) {
    console.error("Feedback Error:", err);
    res.json({ success: false, message: "Error saving feedback" }); }
});
app.get("/feedbacks", async (req, res) => {
  try {
    const feedbacks = await Feedback.find().sort({ createdAt: -1 });
    res.json({ success: true, feedbacks });
  } catch (err) {
    console.error(" Fetch Feedback Error:", err);
    res.json({ success: false, message: "Error fetching feedbacks" }); }
});
```

Screenshots:

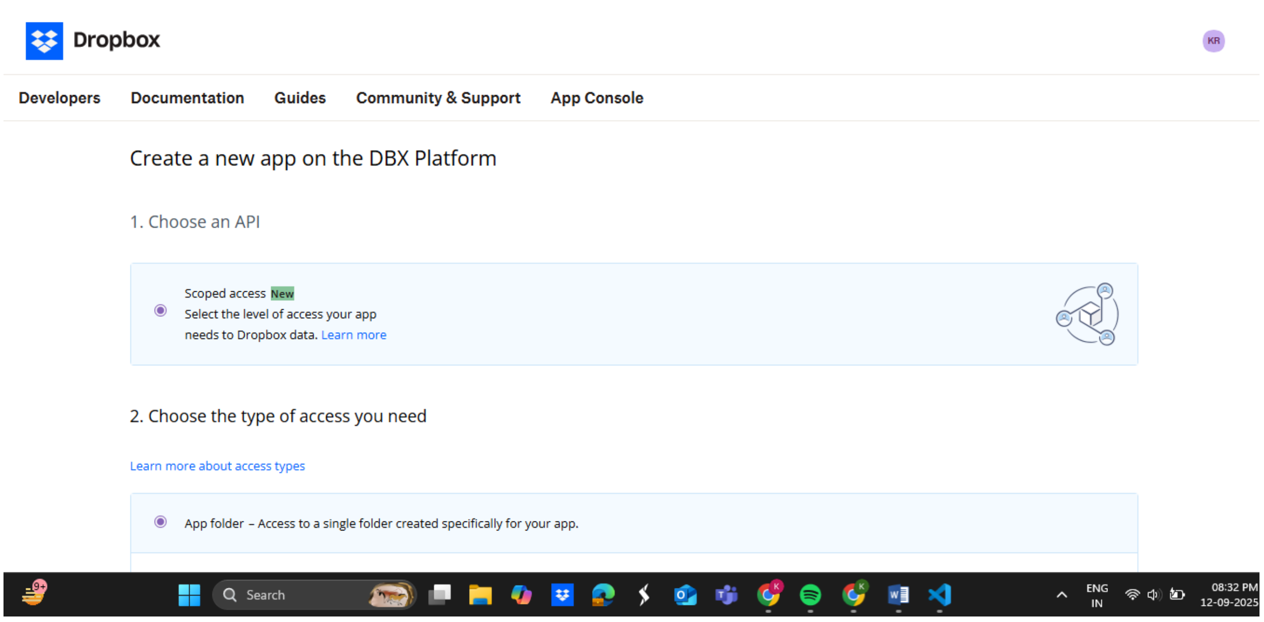


DBX Platform

Develop apps for 700 million Dropbox users

Whether you're building your business or optimizing internal workflows, the DBX Platform lets you add Dropbox features to your apps, such as file storage, sharing, previews, and search.

08:30 PM 12-09-2025



Create a new app on the DBX Platform

1. Choose an API

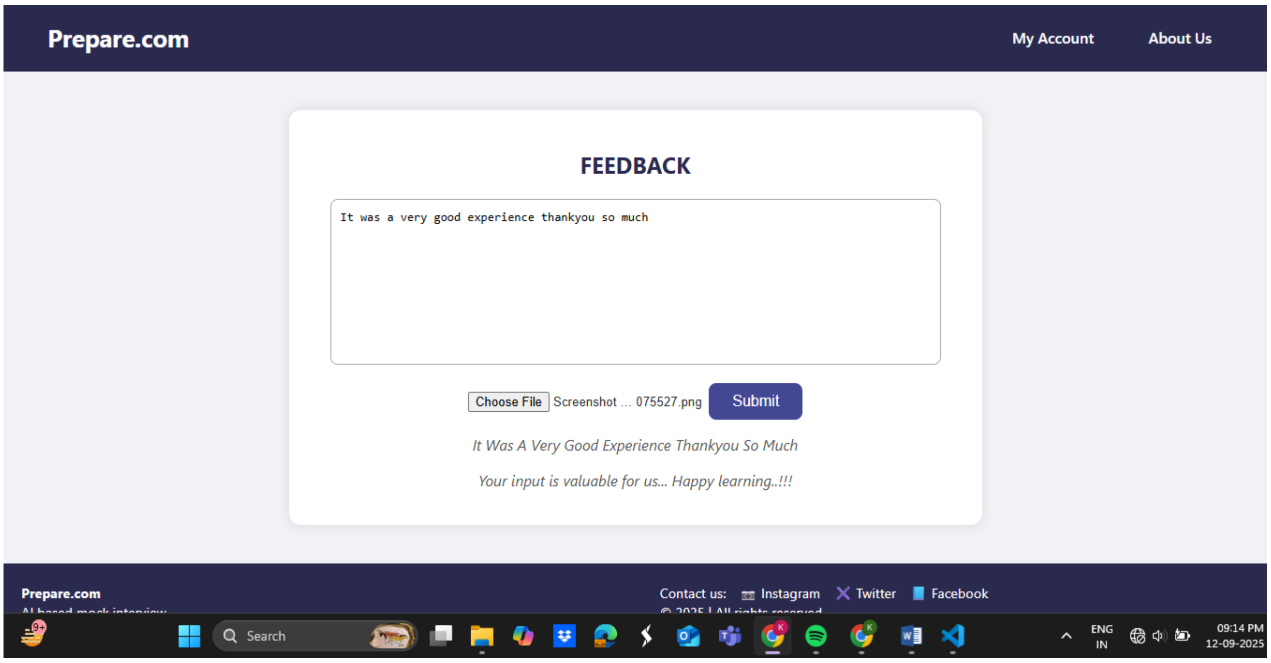
Scoped access New
Select the level of access your app needs to Dropbox data. Learn more

2. Choose the type of access you need

Learn more about access types

App folder – Access to a single folder created specifically for your app.

08:32 PM 12-09-2025



Prepare.com

My Account About Us

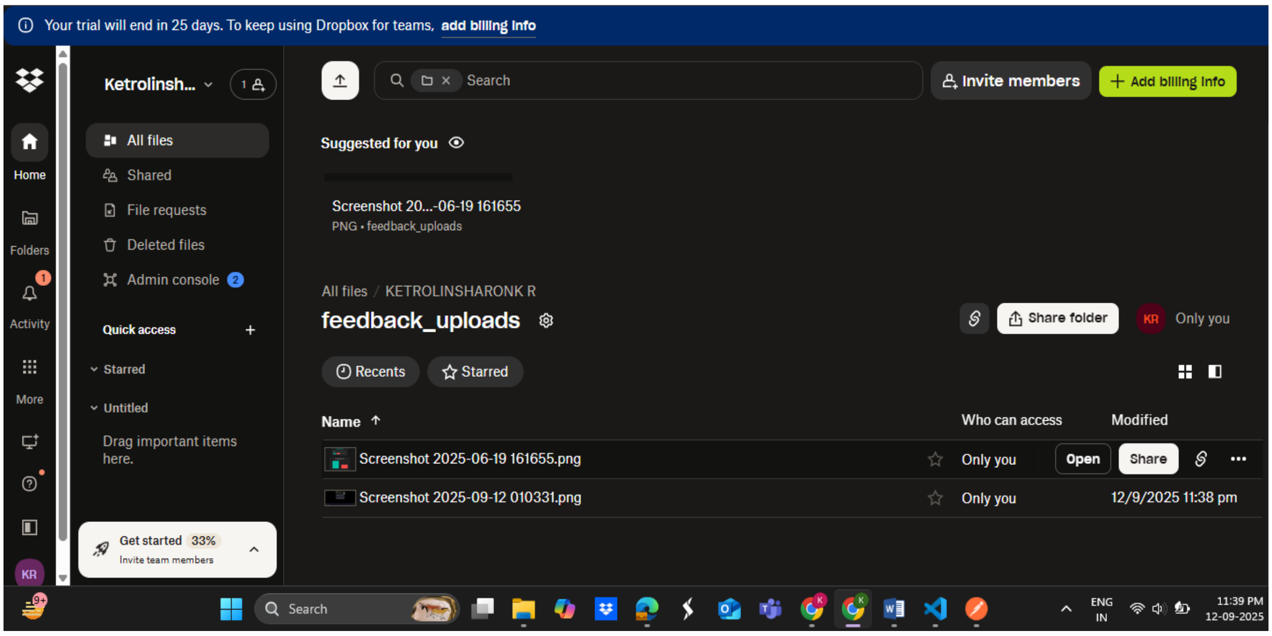
FEEDBACK

```
It was a very good experience thankyou so much
```

Choose File Screenshot ... 075527.png Submit

It Was A Very Good Experience Thankyou So Much

Your input is valuable for us... Happy learning..!!!



Your trial will end in 25 days. To keep using Dropbox for teams, add billing info

Screenshot 20...-06-19 161655
PNG • feedback_uploads

All files / KETROLINSHARONK R

feedback_uploads

Name	Who can access	Modified
Screenshot 2025-06-19 161655.png	Only you	
Screenshot 2025-09-12 010331.png	Only you	12/9/2025 11:38 pm

Result:

Implementation of Dropbox API in our application was done successfully and the results were verified.