**Practical-18**

**Aim:-**

Create Email sending Page which contains To, From, CC, BCC, Subject and   
Message body field.   
Note:   
• Explore How send email with attachments   
• Create and apply email template before sending email  
**Code:-**

**Upload.js:-**

// src/routes/upload.js

const express = require("express");

const multer = require("multer");

const path = require("path");

const router = express.Router();

const fs = require("fs");

// Image Upload

const imageStorage = multer.diskStorage({

  destination: "src/images",

  filename: (req, file, cb) => {

    cb(

      null,

      file.fieldname + "\_" + Date.now() + path.extname(file.originalname)

    );

  },

});

const imageUpload = multer({

  storage: imageStorage,

  limits: { fileSize: 1000000 },

  fileFilter(req, file, cb) {

    if (!file.originalname.match(/\.(png|jpg|JPG)$/)) {

      return cb(new Error("Please upload an Image"));

    }

    cb(null, true);

  },

});

// For Single image upload

router.post(

  "/uploadImage",

  imageUpload.single("image"),

  (req, res) => {

    console.log("Single image uploaded");

    res.send(req.file);

  },

  (error, req, res, next) => {

    console.error("Error uploading single image:", error.message);

    res.status(400).json({ error: error.message });

  }

);

router.get("/getimages", (req, res) => {

  res.sendFile("E:/AWT/Practical File/Pr19/views/getimages.html");

});

router.get("/images", (req, res) => {

  const directoryPath = path.join("src/images");

  fs.readdir(directoryPath, (err, files) => {

    if (err) {

      console.error("Error getting images:", err.message);

      return res

        .status(500)

        .json({ error: `Could not retrieve images ${err}` });

    }

    const images = files.map((file) => `../src/images/${file}`);

    res.json({ images: images });

  });

});

// For Multiple image upload

router.post(

  "/uploadBulkImage",

  imageUpload.array("images", 4),

  (req, res) => {

    console.log("Multiple images uploaded");

    res.send(req.files);

  },

  (error, req, res, next) => {

    console.error("Error uploading multiple images:", error.message);

    res.status(400).json({ error: error.message });

  }

);

// Video Upload

const videoStorage = multer.diskStorage({

  destination: "src/videos",

  filename: (req, file, cb) => {

    cb(

      null,

      file.fieldname + "\_" + Date.now() + path.extname(file.originalname)

    );

  },

});

const videoUpload = multer({

  storage: videoStorage,

  limits: { fileSize: 10000000 },

  fileFilter(req, file, cb) {

    if (!file.originalname.match(/\.(mp4|MPEG-4)$/)) {

      return cb(new Error("Please upload a Video"));

    }

    cb(null, true);

  },

});

router.post(

  "/uploadVideo",

  videoUpload.single("video"),

  (req, res) => {

    console.log("Video uploaded");

    res.send(req.file);

  },

  (error, req, res, next) => {

    console.error("Error uploading video:", error.message);

    res.status(400).json({ error: error.message });

  }

);

router.get("/uploadImage", (req, res) => {

  res.sendFile(path.join(\_\_dirname, "../../views/image.html"));

});

router.get("/uploadBulkImage", (req, res) => {

  res.sendFile(path.join(\_\_dirname, "../../views/multi\_image.html"));

});

router.get("/uploadVideo", (req, res) => {

  res.sendFile(path.join(\_\_dirname, "../../views/video.html"));

});

**app.js:-**

// src/app.js

const express = require("express");

const uploadRoute = require("./routes/upload.js");

const app = express();

const port = process.env.PORT || 7000;

// Middleware for parsing JSON bodies

app.use(express.json());

// Use the upload route

app.use("/uploads", uploadRoute);

// Catch-all route for handling 404 errors

app.use((req, res) => {

  res.status(404).json({ error: "Not Found" });

});

// Start the server

app.listen(port, () => {

  console.log(`Server is up on port ${port}`);

});