GitHub Link: https://github.com/AnujModi13/AWT/tree/main/PracticalList

Practical-18

Aim:-

Create Email sendning 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:-

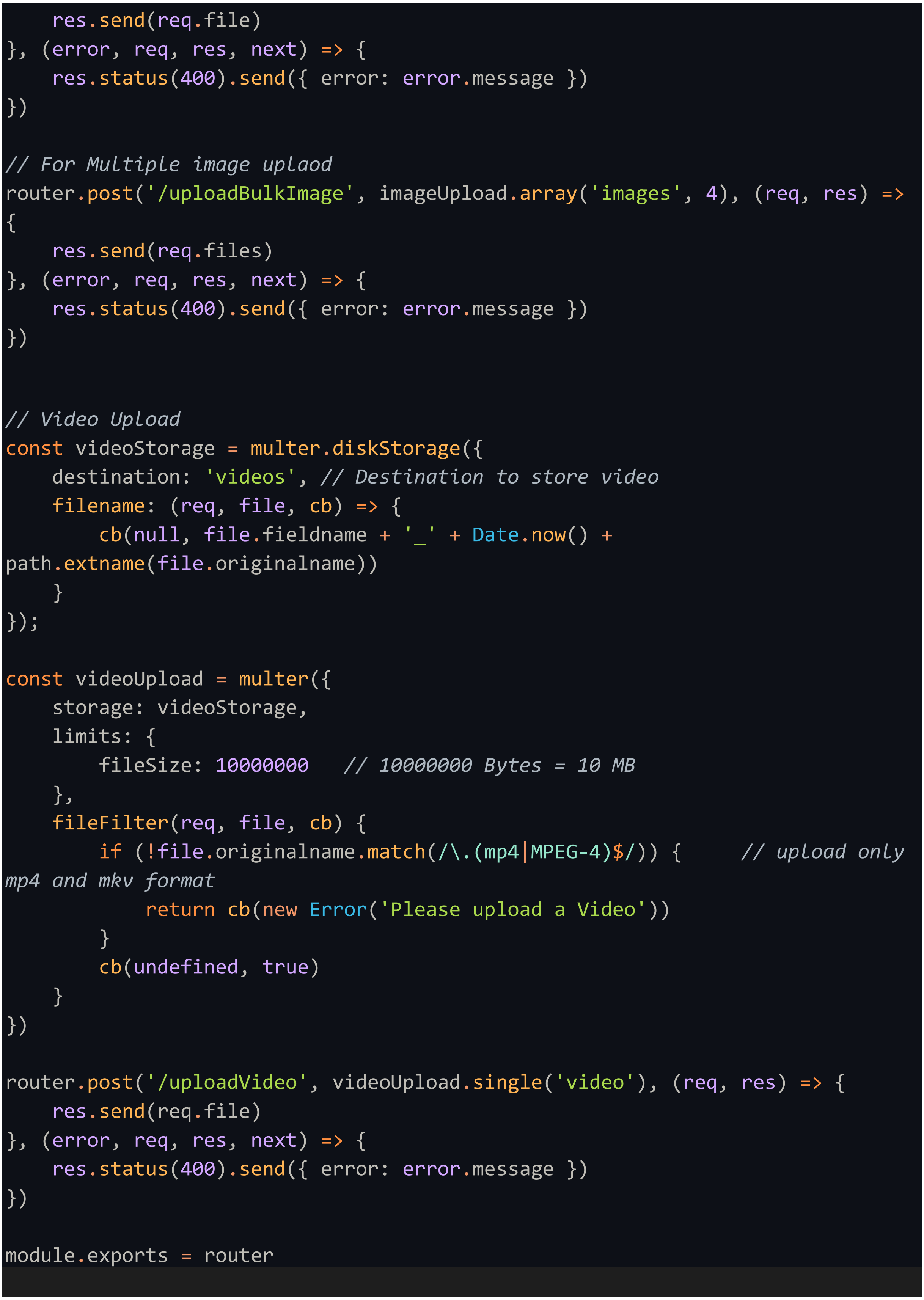
Upload.js:-

```
const express = require('express');
const multer = require('multer');
const path = require('path');
const router = express.Router()

// Image Upload
const imageStorage = multer.diskStorage({
    destination: 'images',
    filename: (req, file, cb) => {
        cb(null, file.fieldname + '_' + Date.now() +
path.extname(file.originalname))
    }
});

const imageUpload = multer({
    storage: imageStorage,
    limits: {
        fileSize: 1000000   // 1000000 Bytes = 1 MB
    },
    fileFilter(req, file, cb) {
        if (!file.originalname.match(/\.(png|jpg|JPG)$/)) {      // upload only
png and jpg format
            return cb(new Error('Please upload a Image'))
        }
        cb(undefined, true)
    }
})

// For Single image upload
router.post('/uploadImage', imageUpload.single('image'), (req, res) => {
```



```
    res.send(req.file)
}, (error, req, res, next) => {
    res.status(400).send({ error: error.message })
})

// For Multiple image uplaod
router.post('/uploadBulkImage', imageUpload.array('images', 4), (req, res) =>
{
    res.send(req.files)
}, (error, req, res, next) => {
    res.status(400).send({ error: error.message })
})


// Video Upload
const videoStorage = multer.diskStorage({
    destination: 'videos', // Destination to store video
    filename: (req, file, cb) => {
        cb(null, file.fieldname + '_' + Date.now() +
path.extname(file.originalname))
    }
});

const videoUpload = multer({
    storage: videoStorage,
    limits: {
        fileSize: 10000000   // 10000000 Bytes = 10 MB
    },
    fileFilter(req, file, cb) {
        if (!file.originalname.match(/\.(mp4|MPEG-4)$/)) {     // upload only
mp4 and mkv format
            return cb(new Error('Please upload a Video'))
        }
        cb(undefined, true)
    }
})

router.post('/uploadVideo', videoUpload.single('video'), (req, res) => {
    res.send(req.file)
}, (error, req, res, next) => {
    res.status(400).send({ error: error.message })
})

module.exports = router
```

Practical 18.js:-

```
const express = require('express');
const uploadRoute = require('./routes/upload');

const app = express()

const port = process.env.PORT || 3000

app.use(uploadRoute);

app.listen(port, () => {
    console.log('Server is up on port ' + port);
})
```