好的，我们来继续完善日记管理功能，增加本地图片上传功能，并参考市面上的日记软件重新设计日记界面的输入项。

**1. 增加本地图片上传功能**

我们需要添加一个用于上传本地图片的表单字段，并使用一个中间件来处理文件上传。

* **安装 multer 中间件：**

npm install multer

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).Bash

* **修改 app.js：**

在 app.js 文件中，引入 multer 中间件，并配置一个用于存储上传文件的目录。 由于代码较长，这里仅列出添加部分，请根据你的实际情况添加到合理的位置：

const multer = require('multer');

// 配置 multer 中间件

const storage = multer.diskStorage({

destination: function (req, file, cb) {

cb(null, 'public/uploads/'); // 设置上传文件的存储目录

},

filename: function (req, file, cb) {

const uniqueSuffix = Date.now() + '-' + Math.round(Math.random() \* 1E9)

cb(null, file.fieldname + '-' + uniqueSuffix + path.extname(file.originalname)); // 设置上传文件的名称

}

});

const upload = multer({ storage: storage });

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript

* **设置 EJS 模板引擎**  
  确保设置了静态页面引用

app.use(express.static('public'));

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript

* **修改 routes/diary.js：**

修改 routes/diary.js 文件，添加 upload.array('images', 5) 中间件，用于处理上传的图片文件。

// 添加日记

app.post('/diary/add', requireLogin,upload.array('images', 5), async (req, res) => {

try {

const { date, title, weather, mood, planList, eventList, feeling, summary, isPublic,location,people,tags } = req.body;

const userId = req.session.userId;

//处理文件名

let imageUrls= [];

if (req.files && req.files.length > 0) {

imageUrls = req.files.map(file => '/uploads/' + file.filename);

}

const newDiary = new Diary({

date: date,

title: title || '无标题',

weather: weather,

mood: mood,

planList: planList ? planList.split('\n') : [], // 支持多行文本

eventList: eventList ? eventList.split('\n') : [], // 支持多行文本

feeling: feeling || '',

summary: summary,

imageUrls: imageUrls, // 图片链接数组

isPublic: isPublic === 'on',

location: location || '',

people: people ? people.split(',') : [],

tags: tags ? tags.split(',') : [],

userId: userId

});

await newDiary.save();

res.redirect('/diary');

} catch (err) {

console.error(err);

res.status(500).send('Server Error');

}

});

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript

* **修改 views/diary/add.ejs 模板：**

修改 views/diary/add.ejs 模板，添加一个用于上传图片的表单字段：

<!DOCTYPE html>

<html>

<head>

<title>添加日记</title>

<!--link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css"-->

<link rel="stylesheet" href="/bootstrap.min.css">

<style>

body {

padding-top: 20px;

}

</style>

</head>

<body>

<div class="container">

<h1>添加日记</h1>

<form action="/diary/add" method="POST" enctype="multipart/form-data"> <!-- 务必添加 enctype="multipart/form-data" -->

<div class="form-group">

<label for="date">日期:</label>

<input type="date" class="form-control" id="date" name="date" required>

</div>

<div class="form-group">

<label for="title">标题:</label>

<input type="text" class="form-control" id="title" name="title">

</div>

<div class="form-group">

<label for="weather">天气:</label>

<input type="text" class="form-control" id="weather" name="weather">

</div>

<div class="form-group">

<label for="mood">心情:</label>

<select class="form-control" id="mood" name="mood">

<option value="">请选择</option>

<option value="happy">快乐</option>

<option value="sad">悲伤</option>

<option value="neutral">平静</option>

<option value="excited">激动</option>

<option value="frustrated">沮丧</option>

</select>

</div>

<div class="form-group">

<label for="location">地点:</label>

<input type="text" class="form-control" id="location" name="location">

</div>

<div class="form-group">

<label for="people">人物:(使用 英文 逗号 , 分割)</label>

<input type="text" class="form-control" id="people" name="people">

</div>

<div class="form-group">

<label for="tags">标签:(使用 英文 逗号 , 分割)</label>

<input type="text" class="form-control" id="tags" name="tags">

</div>

<div class="form-group">

<label for="planList">计划列表:(一行一项)</label>

<textarea class="form-control" id="planList" name="planList" rows="3"></textarea>

</div>

<div class="form-group">

<label for="eventList">事件列表:(一行一项)</label>

<textarea class="form-control" id="eventList" name="eventList" rows="3"></textarea>

</div>

<div class="form-group">

<label for="feeling">心情随笔:</label>

<textarea class="form-control" id="feeling" name="feeling" rows="3"></textarea>

</div>

<div class="form-group">

<label for="summary">每日总结:</label>

<textarea class="form-control" id="summary" name="summary" rows="5" required></textarea>

</div>

<div class="form-group">

<label for="images">图片:</label>

<input type="file" class="form-control" id="images" name="images" multiple accept="image/\*"> <!-- 添加多文件上传 -->

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" id="isPublic" name="isPublic">

<label class="form-check-label" for="isPublic">公开日记</label>

</div>

<button type="submit" class="btn btn-primary">保存</button>

<a href="/diary" class="btn btn-secondary">取消</a>

</form>

</div>

</body>

</html>

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).Html

**2. 参照市面上的日记软件，重新设计日记界面的输入项**

我们来对日记界面的内容进行一个重新的评估与设计，增加一些市面上比较流行的元素到日记页面之中。  
以下是一个重新设计的日志界面的输入项：

* **基本信息:**
  + 日期（必填） - 已经实现
  + 标题 (可选) - 已经实现
  + 天气（可选）
  + 心情 (可选)
  + 地点(可选)
* **丰富内容:**
  + 正文（必填）：支持富文本编辑，例如加粗、斜体、引用、代码块等。
  + 图片上传（可选）：支持多图上传
  + 链接/网址(可选)
  + 语音记录 (可选)
* **辅助信息:**
  + 人物(可选)
  + 标签 (可选)
  + 总结与反思：
  + 计划列表（可选）
  + 事件列表（可选）
  + 心情随笔(可选)
* **其他设置:**
  + 公开/私密设置。

为了达到效果，我们需要进行多项修改， 请按照代码的说明进行修改。  
总体的修改主要围绕模型修改、功能增强以及新增依赖。

1、安装一个新的依赖

npm install quill

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).

为了实现代码富文本， 我们选择直接引用目前流行的方案。 当然你也可以选择其他的富文本方案，只要找到对应的依赖即可。

2、修改/models/diary.js

const mongoose = require('mongoose');

const Schema = mongoose.Schema;

const diarySchema = new mongoose.Schema({

date: { type: Date, required: true },

title: { type: String, default: '无标题' },

weather: { type: String },

mood: { type: String, enum: ['happy', 'sad', 'neutral', 'excited', 'frustrated'] },

location: { type: String },

people: [{ type: String }],

tags: [{ type: String }],

//planList: [{ type: String }], 移除

//eventList: [{ type: String }],

//feeling: { type: String }, 移除

content: { type: String, required: true }, //正文

summary: { type: String }, // 每日总结

imageUrls: [{ type: String }], // 图片 URL 列表

isPublic: { type: Boolean, default: false },

userId: { type: mongoose.Schema.Types.ObjectId, ref: 'User', required: true },

link:{ type: String }, //添加 URL 链接字段

audioUrl: { type: String } // 添加录音链接

});

module.exports = mongoose.model('Diary', diarySchema);

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript

2、创建三个静态资源文件  
这三个文件分别是CSS、JS与html，你需要把三个文件放到pubilc目录下，和之前的bootstap.min.css等文件放在一起。  
这三个文件主要用来实现前端富文本相关的功能

(1) public/style.css

/\* styles.css \*/

#editor {

height: 300px;

}

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript

(2) public/script.js

// public/script.js

var quill = new Quill('#editor', {

theme: 'snow'

});

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript

(3) 引用quill的资源

<link href="https://cdn.quilljs.com/1.3.6/quill.snow.css" rel="stylesheet">

<script src="https://cdn.quilljs.com/1.3.6/quill.js"></script>

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript

3、修改view/diary/add.ejs代码

<!DOCTYPE html>

<html>

<head>

<title>添加日记</title>

<link rel="stylesheet" href="/bootstrap.min.css">

<link href="https://cdn.quilljs.com/1.3.6/quill.snow.css" rel="stylesheet">

<link href="/style.css" rel="stylesheet">

<style>

body {

padding-top: 20px;

}

</style>

</head>

<body>

<div class="container">

<h1>添加日记</h1>

<form action="/diary/add" method="POST" enctype="multipart/form-data">

<div class="form-group">

<label for="date">日期:</label>

<input type="date" class="form-control" id="date" name="date" required>

</div>

<div class="form-group">

<label for="title">标题:</label>

<input type="text" class="form-control" id="title" name="title">

</div>

<div class="form-group">

<label for="weather">天气:</label>

<input type="text" class="form-control" id="weather" name="weather">

</div>

<div class="form-group">

<label for="mood">心情:</label>

<select class="form-control" id="mood" name="mood">

<option value="">请选择</option>

<option value="happy">快乐</option>

<option value="sad">悲伤</option>

<option value="neutral">平静</option>

<option value="excited">激动</option>

<option value="frustrated">沮丧</option>

</select>

</div>

<div class="form-group">

<label for="location">地点:</label>

<input type="text" class="form-control" id="location" name="location">

</div>

<div class="form-group">

<label for="people">人物:(使用 英文 逗号 , 分割)</label>

<input type="text" class="form-control" id="people" name="people">

</div>

<div class="form-group">

<label for="tags">标签:(使用 英文 逗号 , 分割)</label>

<input type="text" class="form-control" id="tags" name="tags">

</div>

<div class="form-group">

<label for="content">内容:</label>

<div id="editor" style="height: 300px;"></div>

<input type="hidden" id="content" name="content">

</div>

<div class="form-group">

<label for="imageUrls">图片:(可多选)</label>

<input type="file" class="form-control" id="images" name="images" multiple accept="image/\*">

</div>

<div class="form-group">

<label for="link">相关链接:</label>

<input type="url" class="form-control" id="link" name="link">

</div>

<div class="form-group">

<label for="summary">每日总结:</label>

<textarea class="form-control" id="summary" name="summary" rows="5" required></textarea>

</div>

<div class="form-check">

<input type="checkbox" class="form-check-input" id="isPublic" name="isPublic">

<label class="form-check-label" for="isPublic">公开日记</label>

</div>

<button type="button" class="btn btn-primary" onclick="handleSubmit()">保存</button>

<a href="/diary" class="btn btn-secondary">取消</a>

</form>

</div>

<script src="https://cdn.quilljs.com/1.3.6/quill.js"></script>

<script src="/script.js"></script>

<script>

function handleSubmit() {

const contentValue = quill.root.innerHTML;

document.getElementById('content').value = contentValue;

document.getElementById('diaryForm').submit();

}

</script>

</body>

</html>

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487" \t "_blank).JavaScript

4、修改routes/diary.js

在做这部分代码修改的时候，需要保证写入日记正文，其他代码不需要修改

app.post('/diary/add', requireLogin,upload.array('images', 5), async (req, res) => {

try {

const { date, title, weather, mood, planList, eventList, feeling, summary, imageUrls, isPublic,location,people,tags,content } = req.body;

const userId = req.session.userId;

//处理文件名

let imageUrls= [];

if (req.files && req.files.length > 0) {

imageUrls = req.files.map(file => '/uploads/' + file.filename);

}

const newDiary = new Diary({

date: date,

title: title || '无标题',

weather: weather,

mood: mood,

planList: planList ? planList.split('\n') : [], // 支持多行文本

eventList: eventList ? eventList.split('\n') : [], // 支持多行文本

feeling: feeling || '',

summary: summary,

imageUrls: imageUrls, // 图片链接数组

isPublic: isPublic === 'on',

location: location || '',

people: people ? people.split(',') : [],

tags: tags ? tags.split(','): [],

content: content,//内容

userId: userId

});

await newDiary.save();

res.redirect('/diary');

} catch (err) {

console.error(err);

res.status(500).send('Server Error');

}

});

content\_copydownload

Use code [with caution](https://support.google.com/legal/answer/13505487).JavaScript

通过上述操作，你应该可以看到一个比较好看和功能强大的日记功能了。  
记得每一次修改都需要进行完整的代码自测，保证各个功能可以正常使用。  
希望能够帮到你。