

重庆大学

学生实验报告

实验课程名称 Web 开发技术

开课实验室 重庆大学 DS1501

学院 大数据与软件学院 年级 2021 专业班 软件工程 X 班

学生姓名 XXX 学号 2021XXXX

开课时间 2023 至 2024 学年第 1 学期

总成绩	
教师签名	XXX

《Web 开发技术》实验报告

开课实验室: DS1501

2023 年 10 月 12 日

学院	大数据与软件学 院	年级、专业、班	2021 级软件工 程 X 班	姓名	XXX	成绩	
课程 名称	Web 开发技术		实验项目 名 称	实验 3: 网页版照片浏览 器的设计与实现		指导教师	XXX
教 师 评 语	<p style="text-align: right;">教师签名: XXX 年 月 日</p>						

一、实验目的（软件需求文档）

1. 实验任务:

设计与实现一个网页版照片浏览器，要求：照片可在后台上传到服务器，在浏览器端，可以按照照片拍摄时间顺序、照片拍摄地点等方式查询照片，也可顺序或倒序翻看所有照片。实现语言不限。提交实验报告文档。

在浏览照片的上方增加查询条件的设置，可以按照照片拍摄的地点和时间进行查询。上载照片的时候不仅仅上载照片图像，还要输入照片的拍摄时间和拍摄地点进行管理。有能力的同学可实现自动获取照片的拍摄时间和拍摄地点进行保存。

2. 实验目的:

本实验旨在让学生通过设计和开发一个自己的网页版照片浏览器页面，掌握以下关键技能和知识：

①上传照片到服务器：搭建一个后台系统，使用户能够将照片上传到服务器。这包括创建一个用户友好的界面，允许用户选择并上传照片。上传的照片将被存储在服务器上以便后续浏览和查询；

②按照时间和地点查询照片：用户应能够按照照片的拍摄时间和拍摄地点进行查询。这意味着需要开发一个查询功能，让用户输入时间范围和/或地点信息，以便筛选照片；

③顺序和倒序翻看照片：提供用户浏览照片的方式，可以按照顺序或倒序查看所有已上传的照片。这通常包括实现一个界面，用户可以单击或滑动来切换照片；

④添加查询条件设置：在照片浏览界面的顶部，应该有一个区域用于设置查询条件。这允许

用户选择拍摄时间范围和地点作为过滤条件，以帮助他们快速找到所需的照片；

⑤输入照片的拍摄时间和地点：用户上传照片时，除了上传照片图像外，还需要输入拍摄时间和拍摄地点的相关信息。这有助于管理和索引照片库，使其更易于查询和组织；

⑥自动获取照片的拍摄时间和地点（可选）：具备能力的同学可以实现一个自动获取照片拍摄时间和地点的功能。这可以通过读取照片的元数据（EXIF 信息）来实现，以便在上传时自动填充相关信息。

总的来说，该实验旨在创建一个功能齐全的照片管理系统，使用户能够上传、浏览和查询照片，同时提供方便的查询条件设置和自动信息提取功能，这有助于更有效地管理和浏览照片集合。

二、实验原理（软件设计文档）

1. 引言

本实验旨在设计和实现一个网页版照片浏览器，这个浏览器允许用户上传照片到服务器，并提供多种方式来浏览和查询照片。这个实验不仅仅关注照片的简单上传和浏览，还强调了照片的管理和查询。用户在上传照片时提供拍摄时间和拍摄地点的信息，这样用户便可以按照照片的拍摄时间和拍摄地点进行查询，还可以按顺序或倒序浏览所有照片。此外，我们还提供了一个查询条件设置界面，使用户可以根据需要轻松筛选照片，以找到他们想要的内容。

2. 前端开发

HTML/CSS/JavaScript：前端开发需要使用这些基本的 Web 技术来创建用户界面，包括照片浏览器界面、上传照片的表单、查询条件设置等。HTML 用于页面结构，CSS 用于样式，JavaScript 用于交互和动态功能。

3. 上传图片到服务器

①HTTP 请求：前端需要使用 HTTP 请求来将用户上传的图片发送到后端服务器。这可能涉及到使用 HTML 表单元素，或通过 JavaScript 使用 AJAX 来发送文件；

②服务器端存储：后端服务器需要接收 HTTP 请求，处理上传的图片，将其存储在服务器上的合适位置，以便后续浏览。

4. 数据库设计

数据库管理系统：使用 MySQL 来存储照片信息，包括拍摄时间和地点等。

5. 查询功能

- ①数据库查询：用户在前端设置查询条件后，前端通过 HTTP 请求将查询条件发送到后端。后端使用数据库查询语句来检索符合条件的照片，并将结果返回给前端；
- ②动态页面更新：前端接收到查询结果后，使用 JavaScript 来动态更新页面，以显示查询结果。

6. 图片元数据处理（可选）

EXIF 信息：如果要实现自动获取拍摄时间和地点的功能，需要使用 JavaScript 或后端代码来解析图片的 EXIF 信息，从中提取拍摄时间和地点。

7. 用户界面设计

- ①响应式设计：设计用户界面时，需要考虑不同屏幕尺寸和设备，以确保网页在各种设备上都能正常显示；
- ②交互设计：创建用户友好的界面，包括照片翻页、查询条件设置、图片上传等。

8. 安全性

- ①文件上传安全：确保在文件上传过程中对文件进行验证和安全处理，以防止恶意文件上传；
- ②数据库安全：实施数据库安全措施，防止 SQL 注入等攻击。

三、使用仪器、材料（软硬件开发环境）

1. 操作系统：Windows 11
2. 开发设备：Lenovo Legion R9000P2021H
3. 开发平台：Eclipse 2022 & IntelliJ IDEA 2022.1.3
4. 测试浏览器：Chrome 浏览器

四、实验步骤（实现的过程）

1. 初始设置

在开始实验之前，确保已经准备好所需的开发工具和环境，包括文本编辑器、浏览器以及根据选择的技术栈准备好的 Java、CSS 和 JSP 文件。

2. 创建 JSP 文件

PictureViewer.jsp:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:f="http://java.sun.com/jsf/core">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Web Picture Viewer</title>
    <link rel="stylesheet" type="text/css" href="PictureViewerStyle.css"/>
</head>
<body>
<f:view>
    <script src="jquery/jquery-3.5.1.js"></script>

    <script>
        <%
            int rsCount; // 记录条数
            String[] picArr;

            // 查询方法
            public boolean search(String sql) {
                final String URL = "jdbc:mysql://localhost:3306/Pictures?useSSL=false&serverTimezone=GMT";
                final String user = "root";
                final String password = "*****";
                Connection con = null;
                Statement stmt = null;
                ResultSet rs = null;

                try {
                    Class.forName("com.mysql.cj.jdbc.Driver");
                    con = DriverManager.getConnection(URL, user, password);

                    stmt = con.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE, ResultSet.CONCUR_READ_ONLY);

                    rs = stmt.executeQuery(sql);

                    int rsnum = 0; // 记录条数

                    rs.last();
                    rsnum = rs.getRow();
                    System.out.println("rsnum:" + rsnum);

                    rsnum = rs.getRow();
                    System.out.println("rsnum:" + rsnum);
                    rs.beforeFirst();
                    if (rsnum == 0) {
                        System.out.println("不存在该照片！");
                        return false;
                    } else {
                        rsCount = rsnum;
                        picArr = new String[rsCount];
                        System.out.println("存在该照片！");
                    }
                } catch (Exception e) {
                    e.printStackTrace();
                } finally {
                    if (stmt != null) {
                        try {
                            stmt.close();
                        } catch (SQLException e) {
                            e.printStackTrace();
                        }
                    }
                    if (con != null) {
                        try {
                            con.close();
                        } catch (SQLException e) {
                            e.printStackTrace();
                        }
                    }
                }
            }
        %>
    </script>
</f:view>
</body>
</html>
```

```
        int num = 0;
        while (rs.next()) {
            System.out.println(rs.getString(1));
            picArr[num] = "src/" + rs.getString(1);
            num++;
        }
        return true;
    }
} catch (ClassNotFoundException e) {
    e.printStackTrace();
} catch (SQLException e) {
    e.printStackTrace();
} catch (Exception e) {
    e.printStackTrace();
} finally {
    // 关闭数据库连接
    if (rs != null)
        try {
            rs.close();
            if (stmt != null)
                stmt.close();
            if (con != null)
                con.close();
        } catch (Exception e1) {
            e1.printStackTrace();
        }
}
return false;
}
%>

<!-- 初始化 -->
<%
    request.setCharacterEncoding("UTF-8");
String Date = request.getParameter("Date");
String Location = request.getParameter("Location");
String sql;

if ((Location == null && Date == null) || (Location.equals("") && Date.equals("")))
{
    sql = "select Name from information";
    System.out.print(sql);
} else if (Location.equals(""))
{
    sql = "select Name from information where Date ='" + Date + "' ;";
} else if (Date.equals(""))
{
    sql = "select Name from information where Location ='" + Location + "' ;";
} else {
    sql = "select Name from information where Date ='" + Date + "' and Location = '" + Location + "' ;";
}
```

```
Boolean isFind = search(sql);
System.out.println(isFind);
%>

if(!<%=isFind%>){
    alert("不存在该照片!");
}
var arr = new Array();
<%
    for (int i = 0; i < rsCount; i++)
{
%>
    arr[<%=i%>] = '<%=picArr[i]>';
    <%
}
%>

var i = 0;
function next(){
    i++;
    if (i == <%=rsCount%>) {
        i = 0;
    }
    document.getElementById("pic").src = arr[i];
}
function last(){
    i--;
    if (i < 0) {
        i = <%=rsCount%> - 1;
    }
    document.getElementById("pic").src = arr[i];
}
```

```
$( '#uploadFile' ).change(function() {
    var form = document.getElementById("uploadF");
    var formData = new FormData(form);
    formData.append('testName', '发送文件标识');
    console.log(formData.get('file'));
    $('#loading').show();
    $.ajax({
        type: "post",
        url: "/WebExp3",
        data: formData,
        processData: false,
        contentType: false,
        success: function(res) {
            console.log(res);
        }
    });
})
```

```

<!-- 照片容器 -->


<div id="pic">
        </img>
    </div>



<!-- 切换照片按钮容器 -->


<input type="button" class="round-button" id="previous-button" value="上一张" onclick="last()">
    <input type="button" class="round-button" id="next-button" value="下一张" onclick="next()">



<!-- 查询照片表单容器 -->


<form action="照片浏览器.jsp" id="search-form" method="post">
        <span class="words">拍摄日期:</span><input type="date" class="text-input" name="Date" />
        <span class="words">拍摄地点:</span><input type="text" class="text-input" name="Location" placeholder="所有地点"/>
        <br/>
        <input type="submit" id="search-button" value="查询照片" />
    </form>



<!-- 上传照片表单容器 -->


<form action="Upload" method="post" id="upload-form" enctype="multipart/form-data">
        <span class="words">选择图片: </span><input name="file" type="file" id="upload-button" value="" /><br/>
        <span class="words">拍摄日期: </span><input name="Date" class="text-input" type="date" value="" /><br/>
        <span class="words">拍摄地点: </span><input name="Location" class="text-input" type="text" value="" />
        <input type="submit" id="submit-button" value="上传" />
    </form>


```

3.添加样式

PictureViewerStyle.css:

```

1  /* 设置整个页面背景图片 */
2  body {
3      background-image: url('src/background.jpg'); /* 背景图片的URL */
4      background-repeat: no-repeat; /* 不重复背景图片 */
5      background-attachment: fixed; /* 固定背景图片 */
6      background-size: 100% 100%; /* 拉伸背景图片以覆盖整个页面 */
7      margin: 0; /* 去除默认的页面边距 */
8      font-family: "微软雅黑"; /* 设置默认字体 */
9  }
10
11 /* 顶部查询容器样式 */
12 .top-container {
13     height: 30px; /* 设置容器高度 */
14 }
15
16 #search-form {
17     background: rgba(0, 0, 0, 0.4); /* 设置容器背景颜色为半透明黑色 */
18     position: absolute; /* 使用绝对定位 */
19     right: 753px; /* 与右边距离753像素 */
20     bottom: 200px; /* 与底部的距离为200像素 */
21 }

```

```
23     #search-button {
24         position: absolute; /* 使用绝对定位 */
25         width: 100px; /* 设置按钮宽度 */
26         height: 30px; /* 设置按钮高度 */
27         border-radius: 5px; /* 设置按钮圆角 */
28         font-size: 18px; /* 设置按钮文本大小 */
29         background: rgba(128, 128, 128, 0.6); /* 设置按钮背景颜色为半透明灰色 */
30         right: 155px; /* 与右边距离155像素 */
31         margin-top: 8px; /* 与顶部的距离为8像素 */
32         color: #FFA500; /* 设置按钮文本颜色为橙色 */
33     }
34
35     #search-button:hover {
36         background-color: #DCDCDC; /* 鼠标悬停时的背景颜色变化 */
37     }
38
39     /* 文本框样式 */
40     .text-input {
41         width: 120px; /* 设置文本框宽度 */
42         height: 30px; /* 设置文本框高度 */
43         text-indent: 5px; /* 文本缩进5像素 */
44         border-radius: 10px; /* 设置文本框圆角 */
45         opacity: 0.75; /* 设置文本框不透明度 */
46         background: rgba(128, 128, 128, 0.5); /* 设置文本框背景颜色为半透明灰色 */
47         color: #FF6347; /* 设置文本框文本颜色为番茄红 */
48     }
49
50     .text-input::placeholder {
51         color: #00FF00; /* 设置文本框占位文本颜色为绿色 */
52     }
53
54     /* 一般文字样式 */
55     .text-label {
56         color: #FFD700; /* 设置文字颜色为金色 */
57         font-size: 18px; /* 设置文字大小 */
58         opacity: 1; /* 设置文字不透明度 */
59     }

```

```
61     /* 照片容器样式 */
62     .photo-container {
63         width: 800px; /* 设置容器宽度 */
64         height: 600px; /* 设置容器高度 */
65         margin: auto; /* 自动居中 */
66         margin-top: 5px; /* 与顶部的距离为5像素 */
67         background-color: #c2c5c5; /* 设置容器背景颜色 */
68         border: 5px inset rgba(128, 128, 128, 0.9); /* 设置容器边框 */
69         border-radius: 10px; /* 设置容器圆角 */
70         position: relative; /* 使用相对定位 */
71         background: rgb(191, 194, 193); /* 设置容器背景颜色 */
72     }

```

```
74     #photo {
75         position: absolute; /* 使用绝对定位 */
76         width: 100%; /* 设置图片宽度为容器宽度的百分之百 */
77         height: 100%; /* 设置图片高度为容器高度的百分之百 */
78         object-fit: scale-down; /* 图片自适应容器 */
79     }
80
81     /* 图片按钮容器样式 */
82     .button-container {
83         width: 800px; /* 设置容器宽度 */
84         margin: auto; /* 自动居中 */
85         position: relative; /* 使用相对定位 */
86     }
87
88     .action-button {
89         width: 305px; /* 设置按钮宽度 */
90         padding: 6px; /* 设置按钮内边距 */
91         border-radius: 10px; /* 设置按钮圆角 */
92         font-size: 20px; /* 设置按钮文本大小 */
93         background: rgba(211, 211, 211, 0.6); /* 设置按钮背景颜色为半透明灰色 */
94         font-family: "微软雅黑"; /* 设置按钮字体 */
95     }
96
97     .action-button:hover {
98         background-color: #DCDCDC; /* 鼠标悬停时的背景颜色变化 */
99     }
100
101    #previous-button {
102        position: absolute; /* 使用绝对定位 */
103        top: 5px; /* 与顶部的距离为5像素 */
104        left: 0; /* 设置left为0，将按钮放到照片容器的左边 */
105        font-weight: 900; /* 设置按钮文本加粗 */
106    }
```

```
108    #next-button {
109        position: absolute; /* 使用绝对定位 */
110        top: 5px; /* 与顶部的距离为5像素 */
111        right: 0; /* 设置right为0，将按钮放到照片容器的右边 */
112        font-weight: 900; /* 设置按钮文本加粗 */
113    }
114
115    .round-button {
116        width: 50px; /* 设置按钮宽度 */
117        height: 50px; /* 设置按钮高度 */
118        border-radius: 50%; /* 将边框半径设置为按钮宽度的一半，创建圆形按钮 */
119        background-color: #d6eae6; /* 设置按钮背景颜色 */
120        color: #40f8d4; /* 设置文本颜色为蓝色 */
121        text-align: center; /* 文本水平居中 */
122        line-height: 50px; /* 文本垂直居中 */
123        cursor: pointer; /* 显示鼠标指针 */
124    }
```

```
126     /* 上传容器样式 */
127     #upload-container {
128         background: #7b9e6d; /* 设置容器背景颜色 */
129         width: 330px; /* 设置容器宽度 */
130         position: absolute; /* 使用绝对定位 */
131         bottom: 0; /* 设置底部为0，将容器放到页面底部 */
132         left: 50%; /* 使用左侧的50%作为定位基准 */
133         transform: translateX(-50%); /* 使用transform属性将容器向左平移自身宽度的一半，实现水平居中 */
134     }
135
136     #upload-form {
137         margin-top: 13px; /* 与顶部的距离为13像素 */
138     }
139
140     #upload-button::file-selector-button {
141         width: 100px; /* 设置按钮宽度 */
142         height: 30px; /* 设置按钮高度 */
143         border-radius: 5px; /* 设置按钮圆角 */
144         font-size: 18px; /* 设置按钮文本大小 */
145         background: #FF69B4; /* 设置上传按钮背景颜色为热情粉红 */
146         font-family: "微软雅黑"; /* 设置按钮字体 */
147         color: #FFFF00; /* 设置上传按钮文本颜色为黄色 */
148     }
149
150     #upload-button:hover {
151         background-color: #DCDCDC; /* 鼠标悬停时的背景颜色变化 */
152     }
153
154     #submit-button {
155         width: 70px; /* 设置按钮宽度 */
156         border-radius: 5px; /* 设置按钮圆角 */
157         font-size: 20px; /* 设置按钮文本大小 */
158         background: #00FFFF; /* 设置上传按钮背景颜色为青色 */
159         font-family: "微软雅黑"; /* 设置按钮字体 */
160         color: #8A2BE2; /* 设置上传按钮文本颜色为蓝紫色 */
161     }
162
163     #submit-button:hover {
164         background-color: #DCDCDC; /* 鼠标悬停时的背景颜色变化 */
165     }
```

4.Java Servlet编程

PicServlet.java:

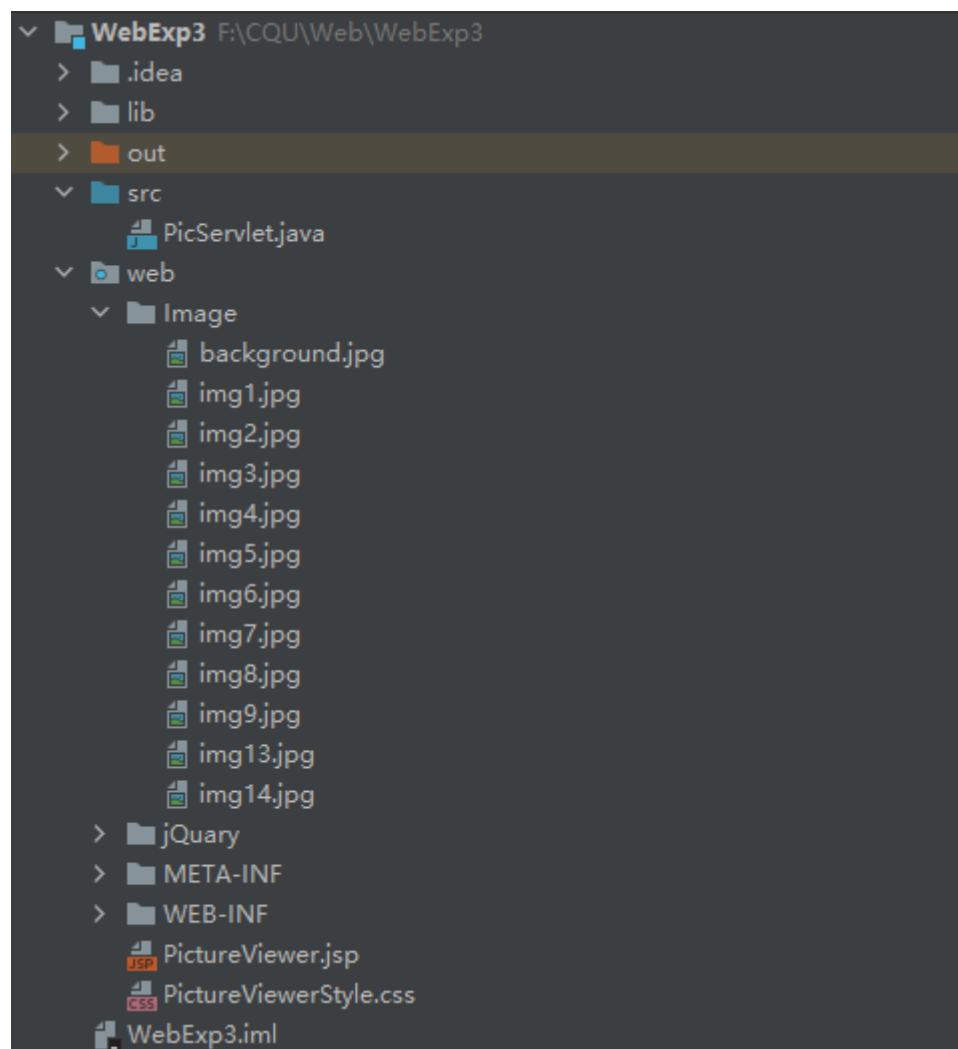
```
1 import java.io.File;
2 import java.io.IOException;
3 import java.io.PrintWriter;
4 import java.util.List;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 import org.apache.commons.fileupload.FileItem;
12 import org.apache.commons.fileupload.FileItemFactory;
13 import org.apache.commons.fileupload.FileUploadException;
14 import org.apache.commons.fileupload.disk.DiskFileItemFactory;
15 import org.apache.commons.fileupload.servlet.ServletFileUpload;
16 import java.sql.Connection;
17 import java.sql.DriverManager;
18 import java.sql.PreparedStatement;
19
20 public class PicServlet extends HttpServlet {
21     private static final long serialVersionUID = 1L;
22
23     public PicServlet() {
24         super();
25     }
26
27     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
28         response.getWriter().append("Served at: ").append(request.getContextPath());
29     }
30
31     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
32         // 设置请求和响应的字符编码
33         request.setCharacterEncoding("UTF-8");
34         response.setCharacterEncoding("UTF-8");
35         response.setContentType("text/html; charset=UTF-8");
36
37         // 检查请求是否包含文件上传内容
38         boolean isMultipart = ServletFileUpload.isMultipartContent(request);
39         if (isMultipart) {
40             FileItemFactory factory = new DiskFileItemFactory();
41             ServletFileUpload upload = new ServletFileUpload(factory);
42             try {
43                 // 解析请求，获取文件项列表
44                 List<FileItem> items = upload.parseRequest(request);
45
46                 // 获取上传的文件项
47                 FileItem upLoadFile = items.get(0);
48
49                 // 获取上传的文件名
50                 String filename = upLoadFile.getName();
51
52                 // 设置文件上传路径，根据需要修改
53                 String path = "F:\\CQU\\Web\\WebExp3\\src\\Image";
54                 File file = new File(path, filename);
55
56                 // 将上传的文件保存到指定路径
57                 upLoadFile.write(file);
58
59                 // 打印上传成功的消息
60                 System.out.println(filename + " OK!!!");
61             } catch (Exception e) {
62                 e.printStackTrace();
63             }
64         }
65     }
66 }
```

```
62 // 从表单获取拍摄日期
63 String time = items.get(1).getString("UTF-8");
64
65 // 从表单获取拍摄地点
66 String location = items.get(2).getString("UTF-8");
67
68 // 插入文件信息到数据库，并获取插入是否成功
69 boolean isOk = fill(filename, time, location);
70
71 // 获取响应输出流
72 PrintWriter pw = response.getWriter();
73
74 if (isOk) {
75     // 如果插入成功，显示上传成功消息，然后跳转到照片浏览器页面
76     pw.print("<script language='javascript'>alert('上传成功');window.location.href='照片浏览器.jsp';</script>");
77 } else {
78     // 如果插入失败，显示上传失败消息，然后跳转到照片浏览器页面
79     pw.print("<script language='javascript'>alert('上传失败');window.location.href='照片浏览器.jsp';</script>");
80 }
81
82 } catch (FileUploadException e1) {
83     e1.printStackTrace();
84 } catch (Exception e2) {
85     e2.printStackTrace();
86 }
87 }
88 }

90 public boolean fill(String filename, String date, String location) {
91     // 设置数据库连接参数
92     final String URL = "jdbc:mysql://localhost:3306/Pictures?useSSL=false&serverTimezone=GMT";
93     final String user = "root";
94     final String password = "021011";
95     Connection con = null;
96     PreparedStatement pstmt = null;
97
98     // 插入数据的SQL语句
99     String sql = "INSERT INTO information(Name, Date, Location) VALUES (?, ?, ?)";
100
101 try {
102     // 加载MySQL数据库驱动
103     Class.forName(className: "com.mysql.cj.jdbc.Driver");
104
105     // 建立数据库连接
106     con = DriverManager.getConnection(URL, user, password);
107     pstmt = con.prepareStatement(sql);
108
109     // 设置SQL语句的参数
110     pstmt.setString(parameterIndex: 1, filename);
111     pstmt.setString(parameterIndex: 2, date);
112     pstmt.setString(parameterIndex: 3, location);
113
114     // 执行SQL语句，返回受影响的行数
115     int ok = pstmt.executeUpdate();
116
117     // 如果受影响的行数大于0，插入成功
118     return ok > 0;
119 } catch (ClassNotFoundException | IOException e) {
120     e.printStackTrace();
121 } catch (Exception e) {
122     e.printStackTrace();
123 } finally {
124     // 关闭数据库连接
125     try {
126         if (pstmt != null) {
127             pstmt.close();
128         }
129         if (con != null) {
130             con.close();
131         }
132     }
133 }
```

```
132     } catch (Exception e) {  
133         e.printStackTrace();  
134     }  
135 }  
136 // 默认返回插入失败  
137 return false;  
138 }  
139 }
```

5. 整体结构



6. 测试和调试

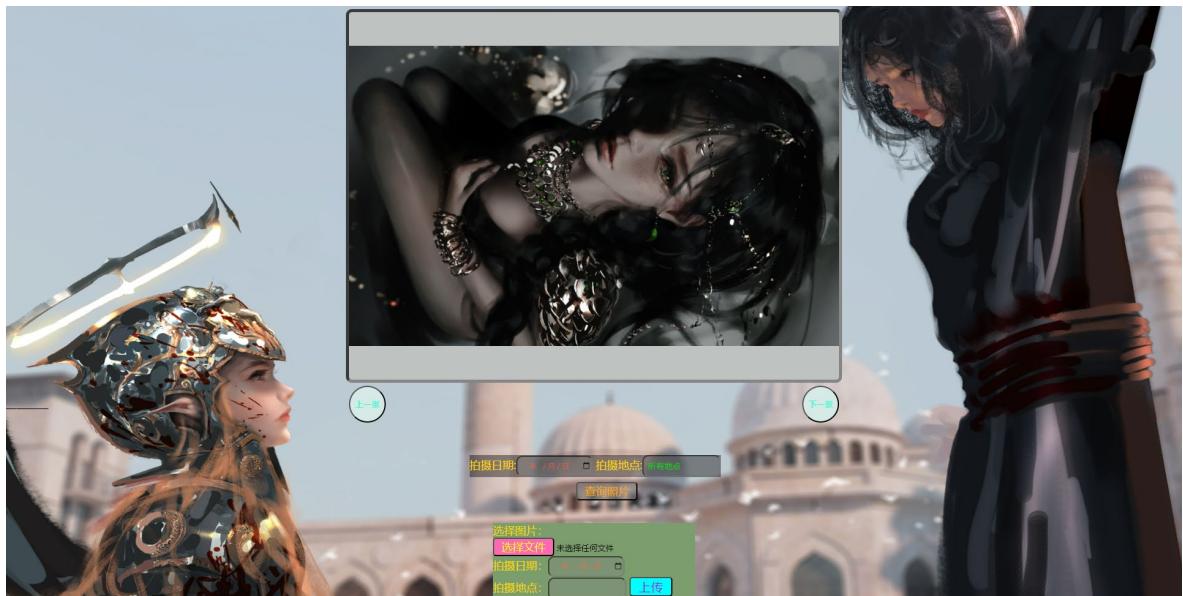
- ①在Chrome浏览器中打开该照片浏览器，检查是否能正常上传、浏览图片；
- ②进行测试，确保照片浏览器在不同屏幕尺寸和设备上具有良好的响应性。

五、实验结果及分析（实现的效果，包括屏幕截图、系统总体运行情况和测试情况等）

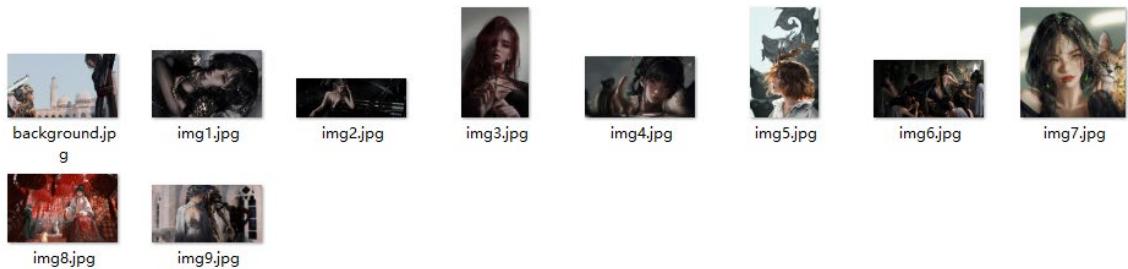
1. 实现效果：

(1) Web 主页面

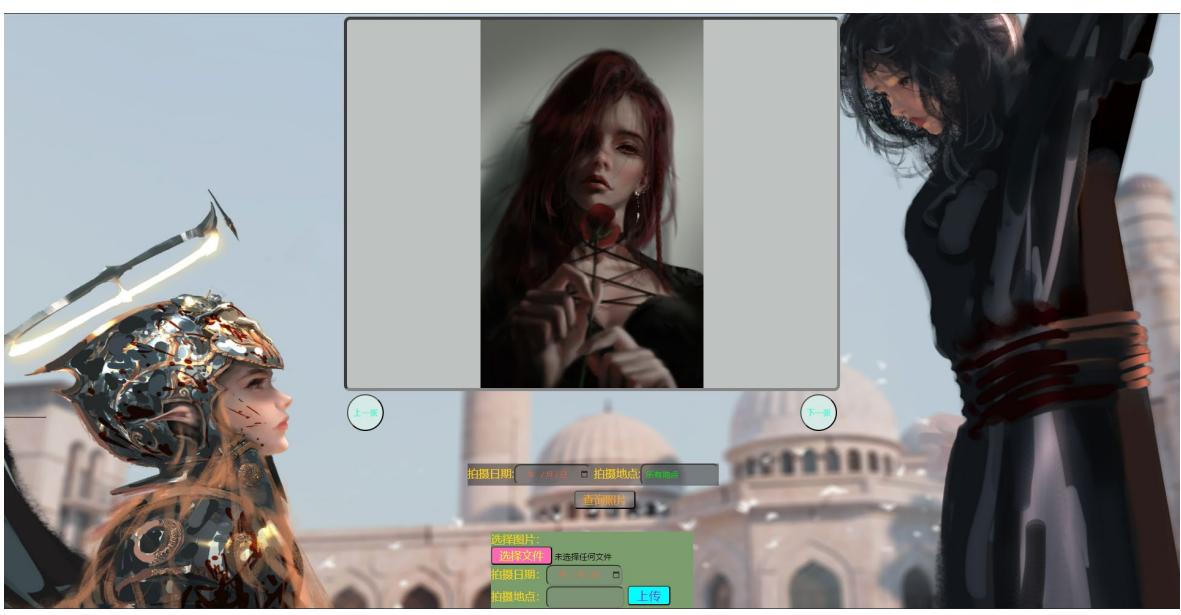
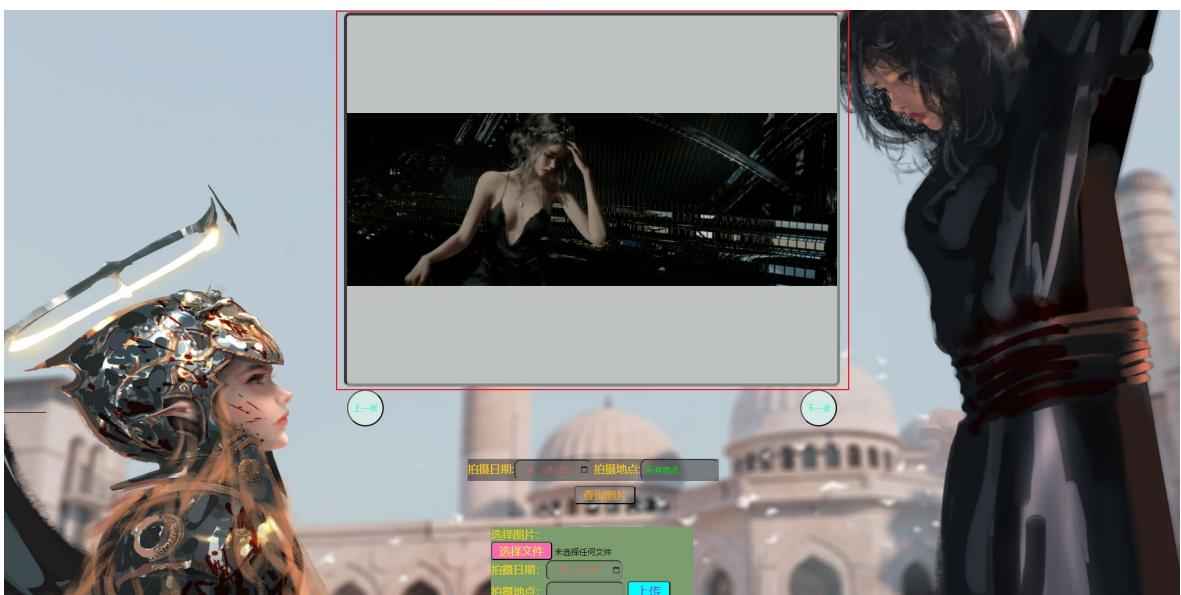
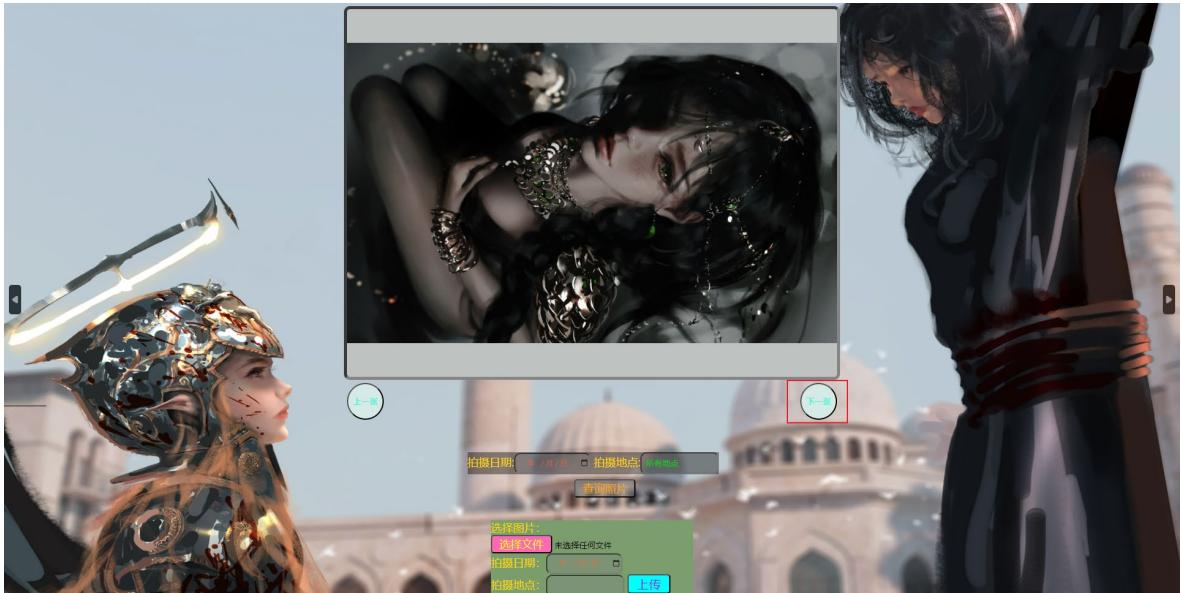
运行程序可以得到如下结果，默认显示第一张图片，图片下方是“上一张”“下一张”切换按钮、分类查询模块以及上传图片模块。



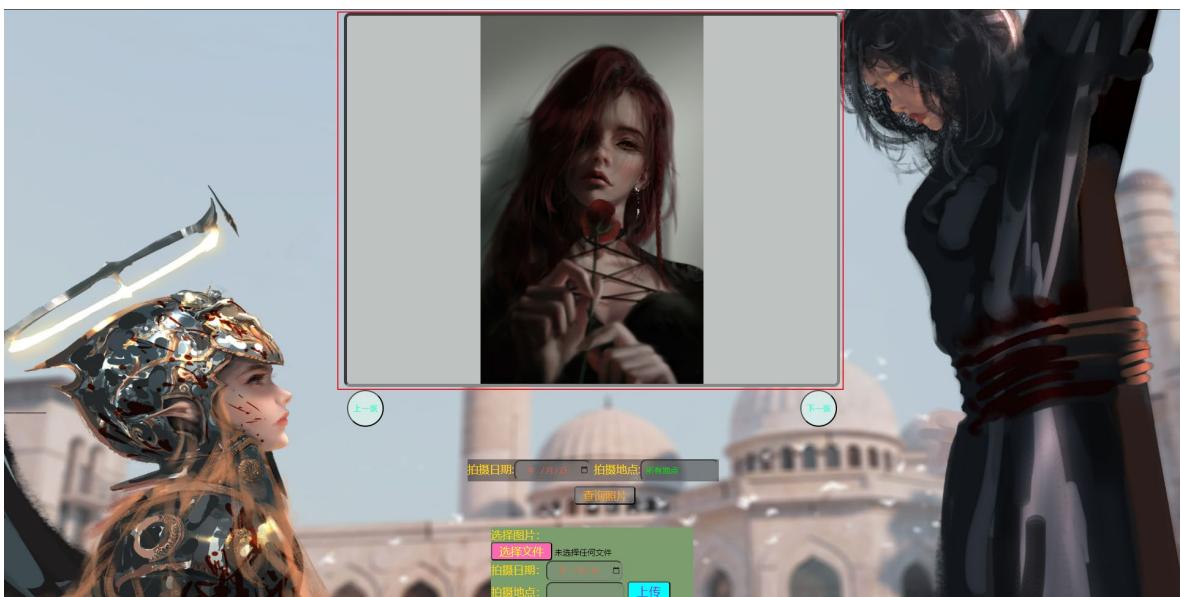
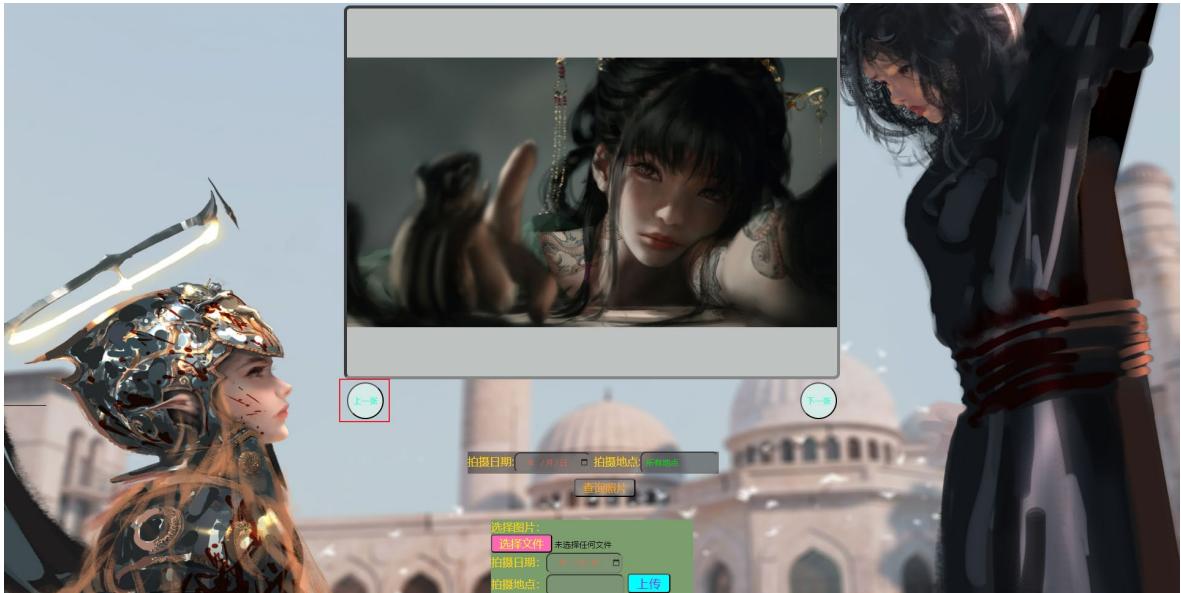
(2) 服务器已存储图片如下；



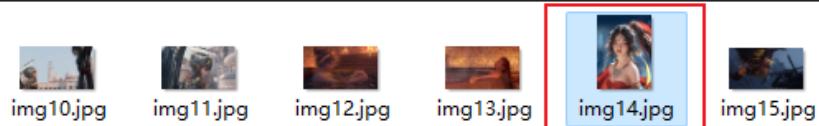
(3) 点击“下一张”可以切换到下一张照片进行浏览，实现正序浏览功能；

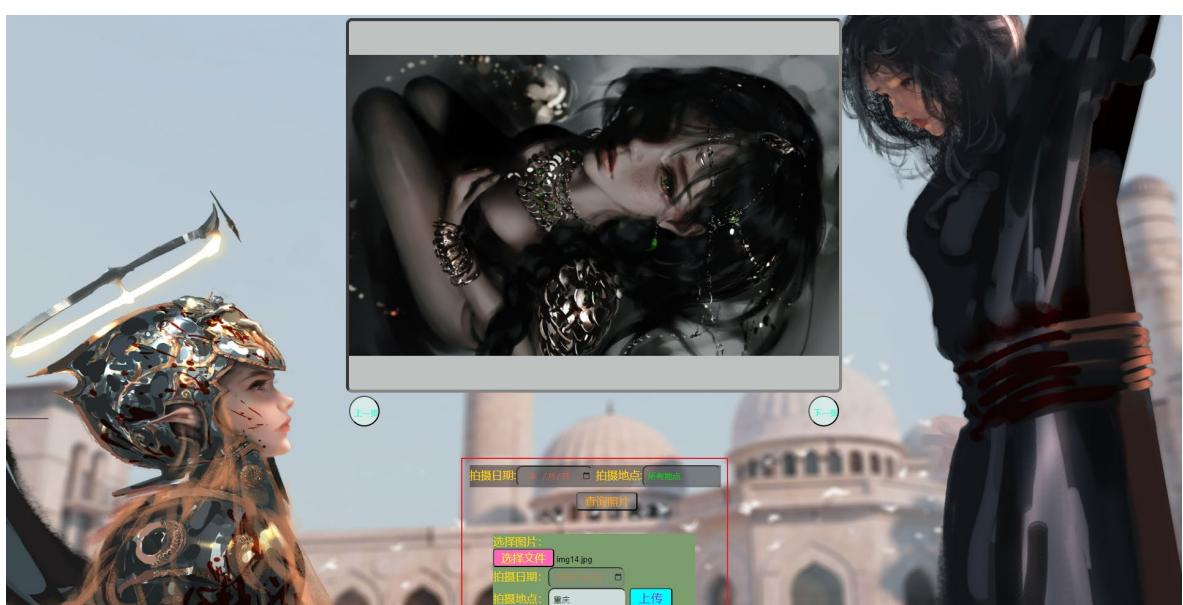


(4) 点击“上一张”可以切换到下一张照片进行浏览，实现逆序浏览功能；



(5) 上传图片并输入信息；

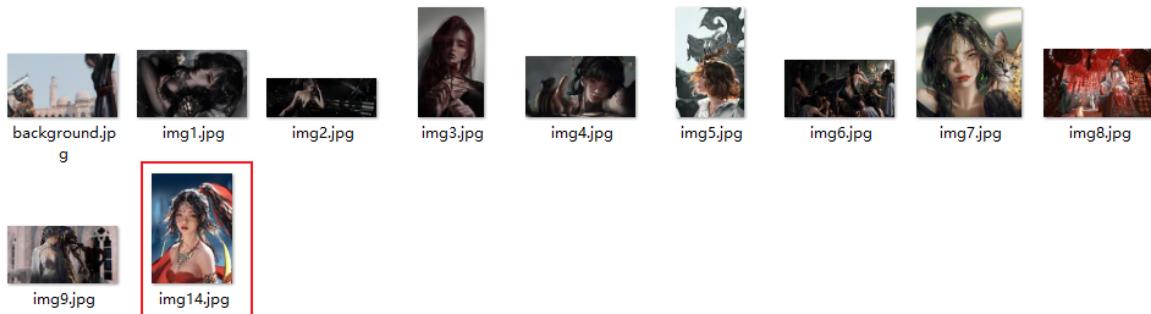




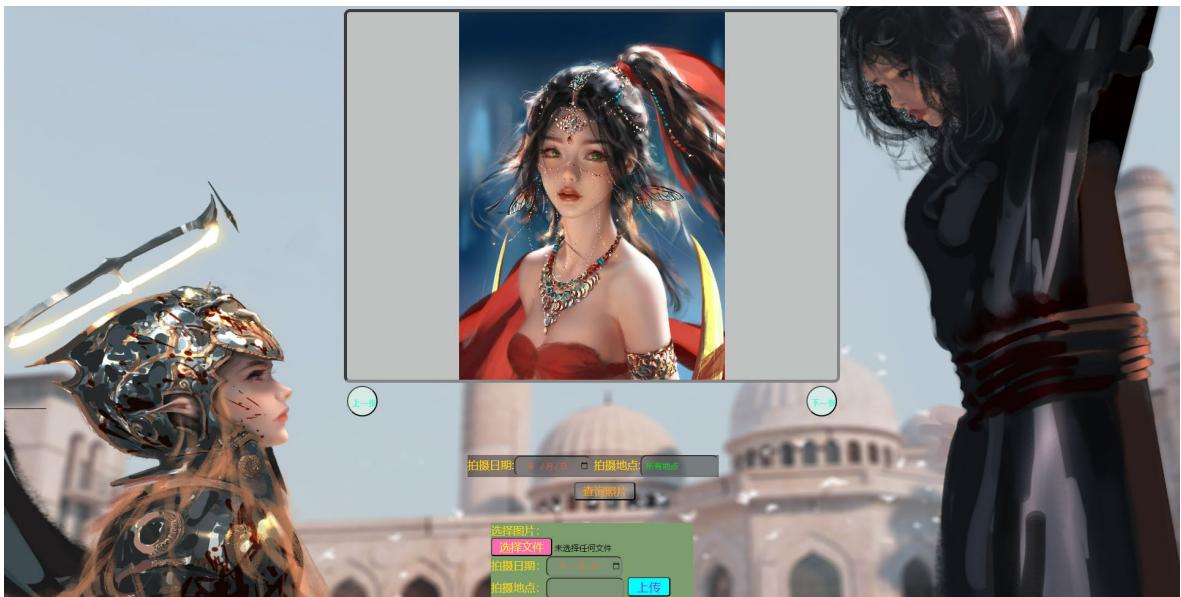
上传成功提示:



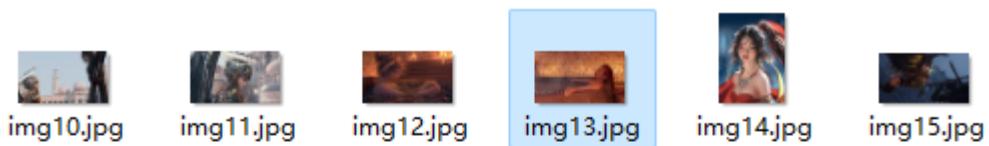
可在服务器端文件夹中检查到成功上传结果:



可在主页面浏览到刚上传图片：



(6) 上传图片时若不输入信息，默认拍摄日期与地点为 NULL；



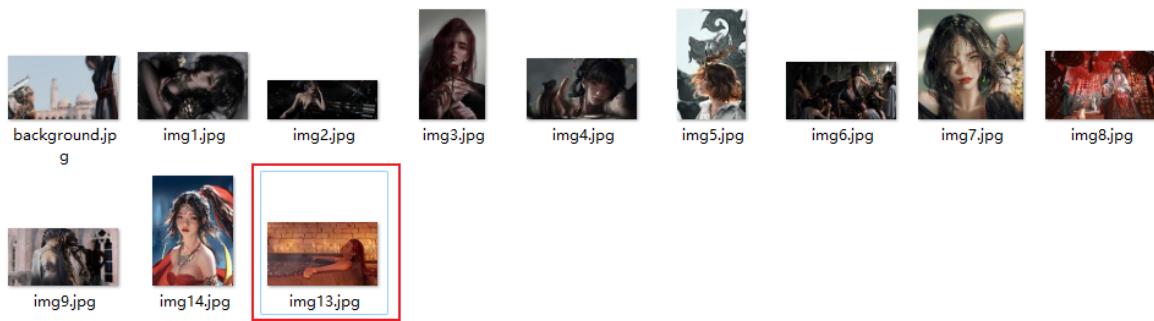
上传成功提示：

localhost:8080 显示

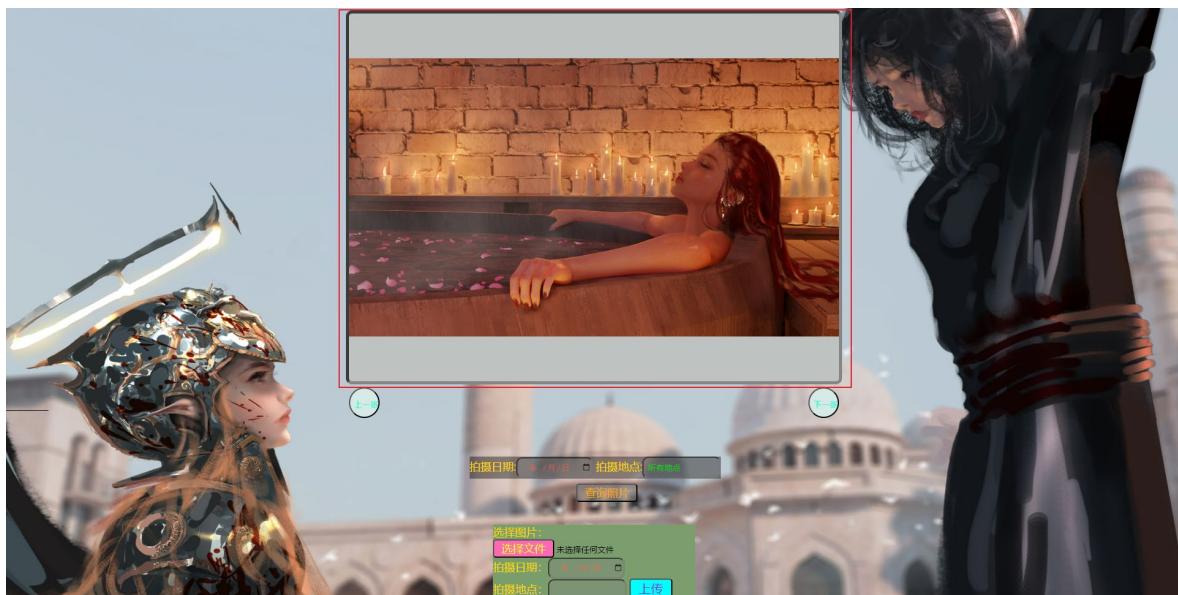
上传成功！

确定

可在服务器端文件夹中检查到成功上传结果：



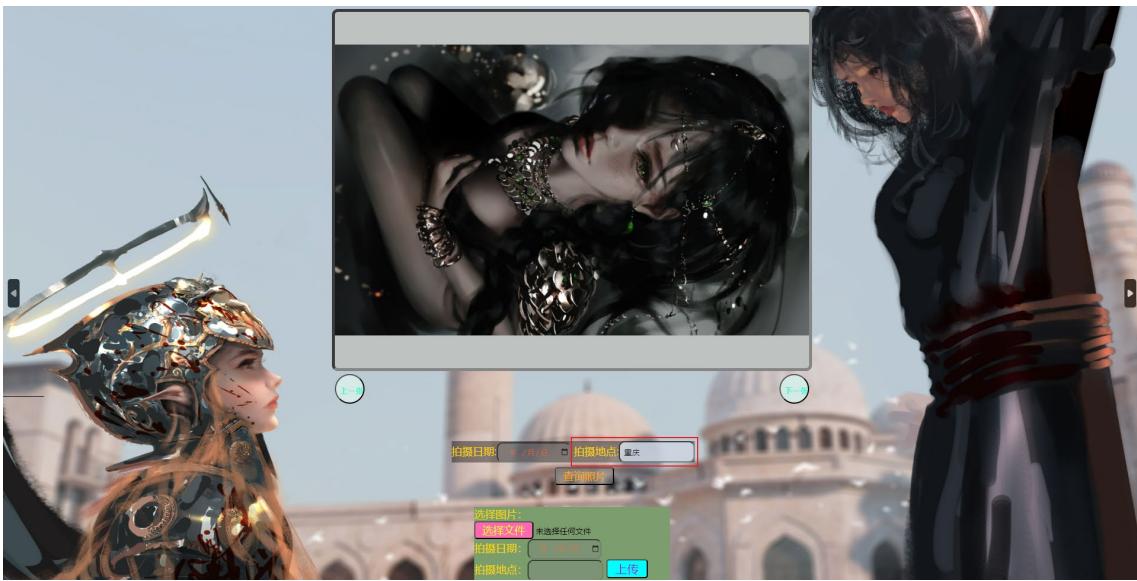
可在主页面浏览到刚上传图片：



数据库中存储信息为：

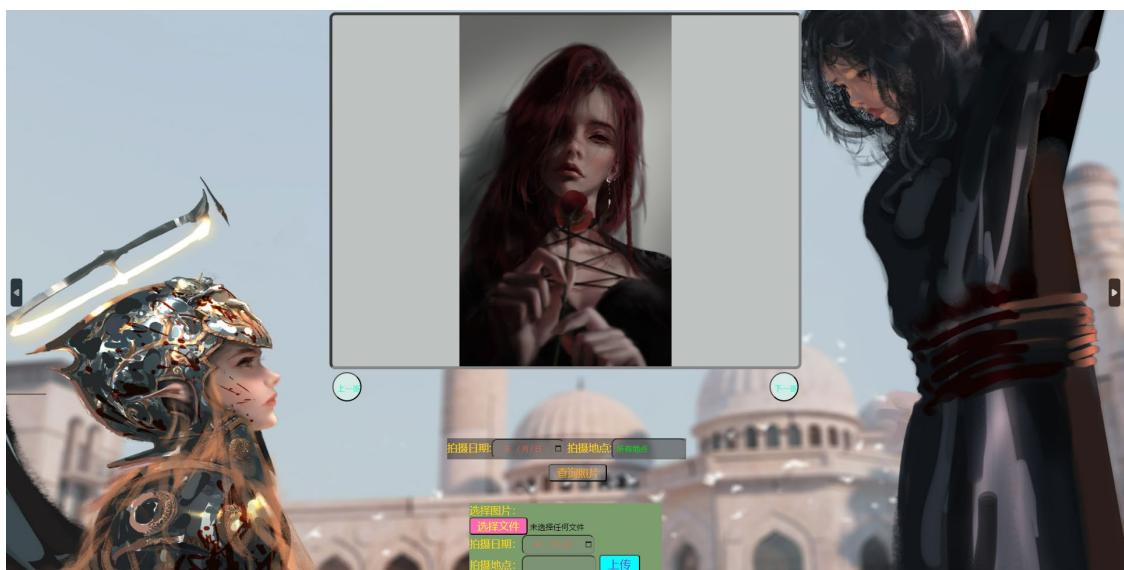
1	img1.jpg	纽约	2023-10-19
2	img2.jpg	天津	2023-10-19
3	img3.jpg	重庆	2023-10-19
4	img4.jpg	上海	2023-10-20
5	img5.jpg	北京	2023-10-20
6	img6.jpg	南京	2023-10-20
7	img7.jpg	重庆	2023-10-21
8	img8.jpg	西藏	2023-10-21
9	img9.jpg	南京	2023-10-21
10	img14.jpg	重庆	2023-10-20
11	img13.jpg	NULL	NULL

(7) 按地点进行查询

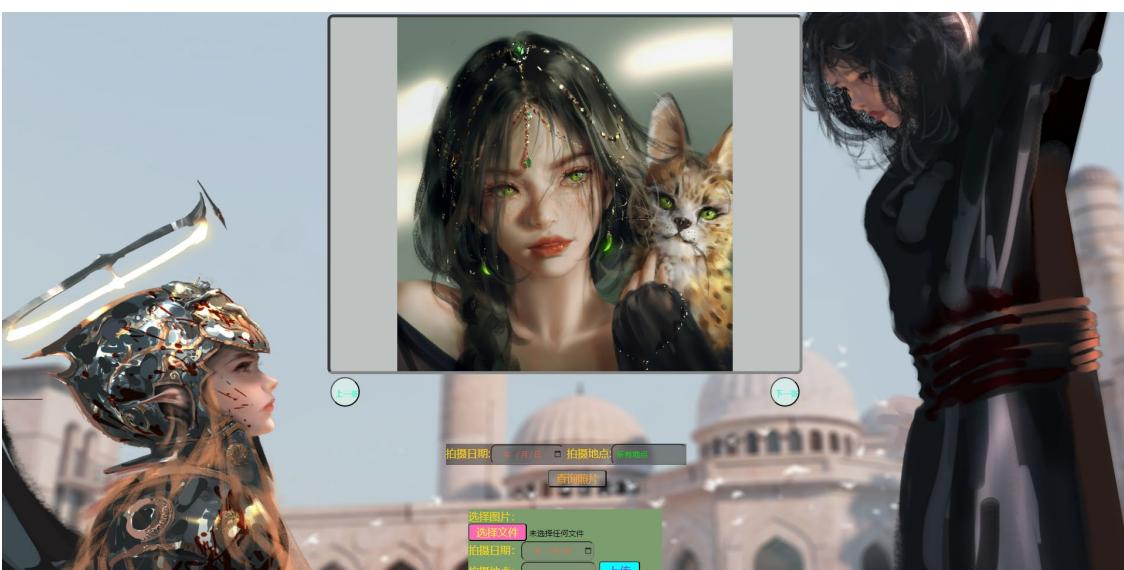


得到结果：

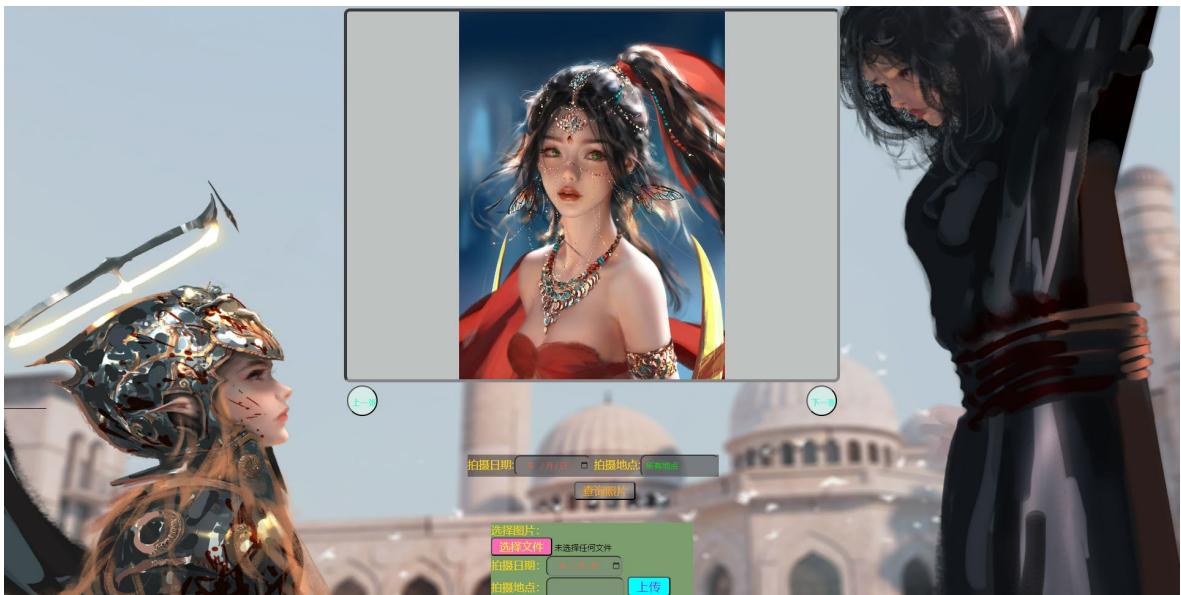
①



②

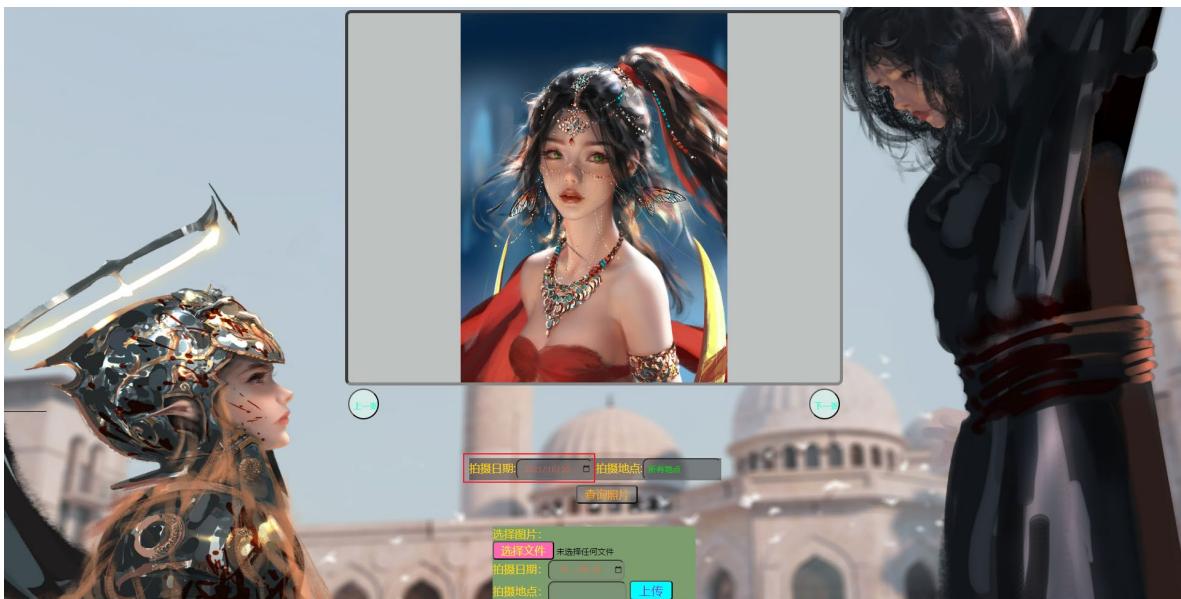


(3)



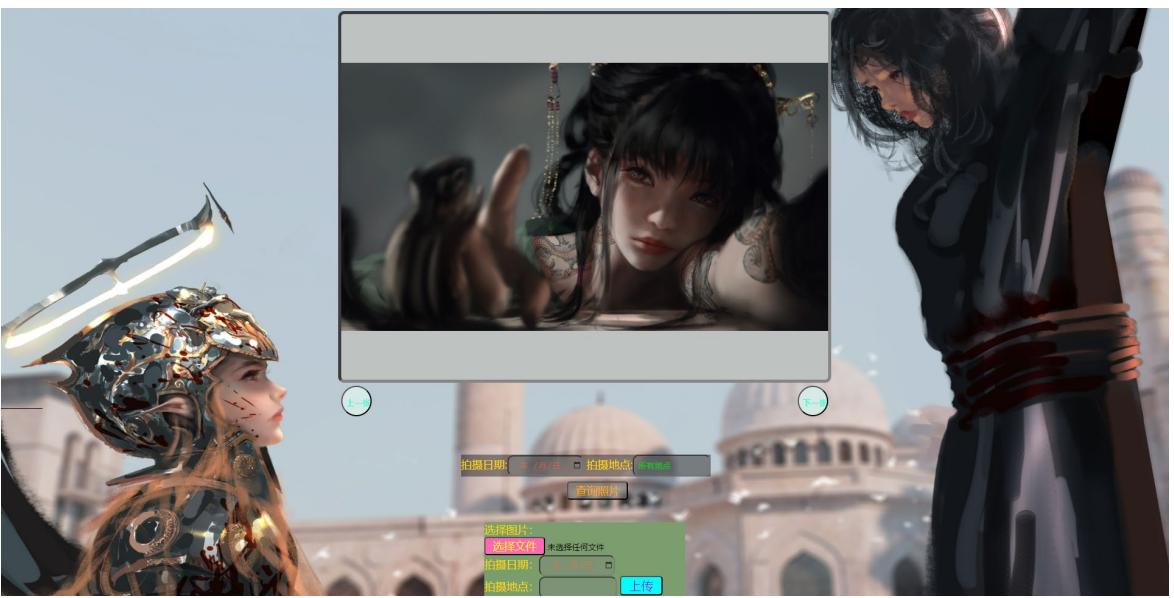
得到结果符合已存储图片情况，正确无误。

(8) 按日期进行查询

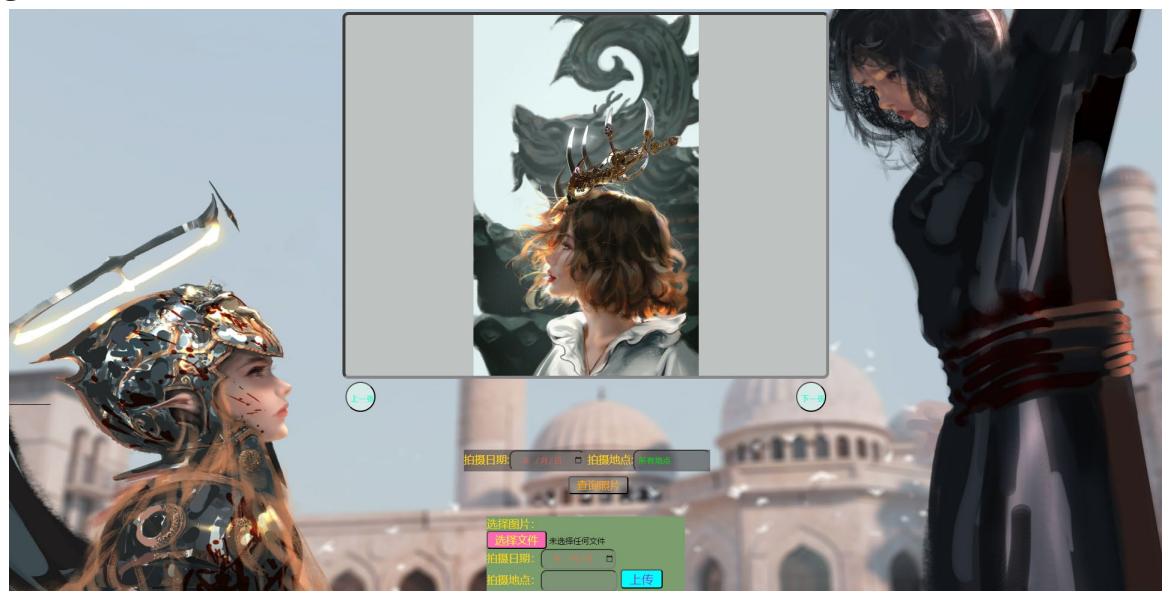


得到结果：

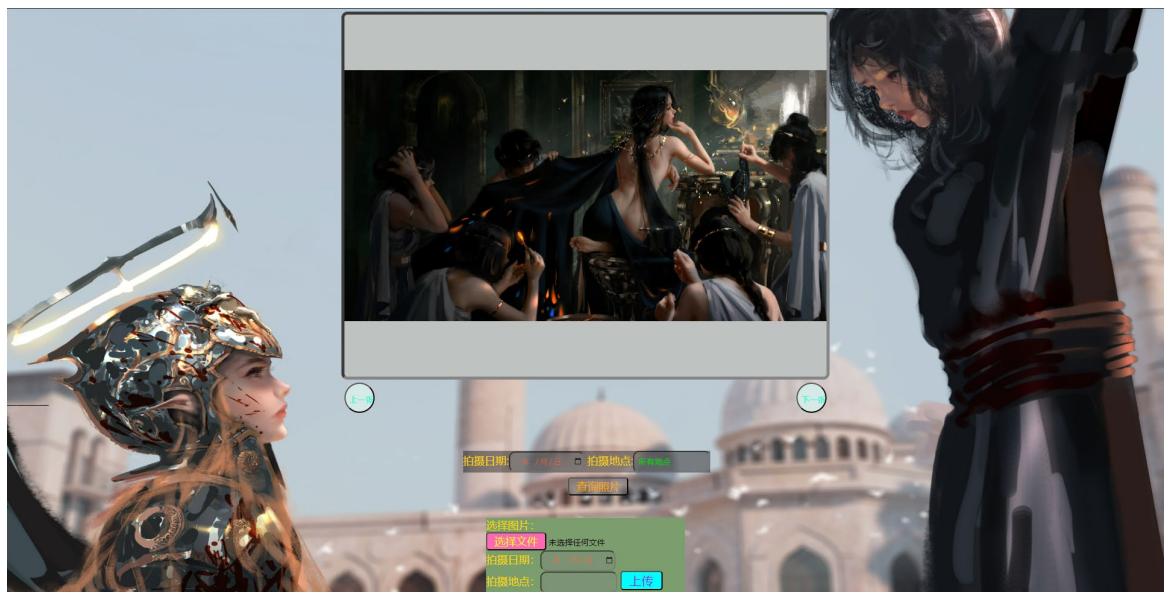
①



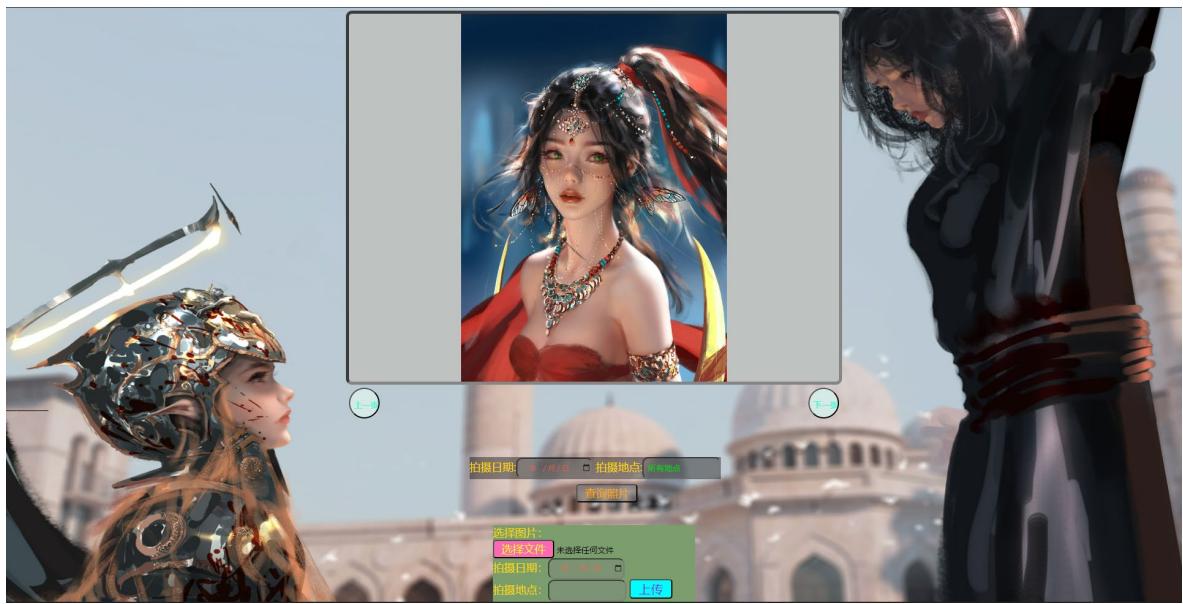
②



③

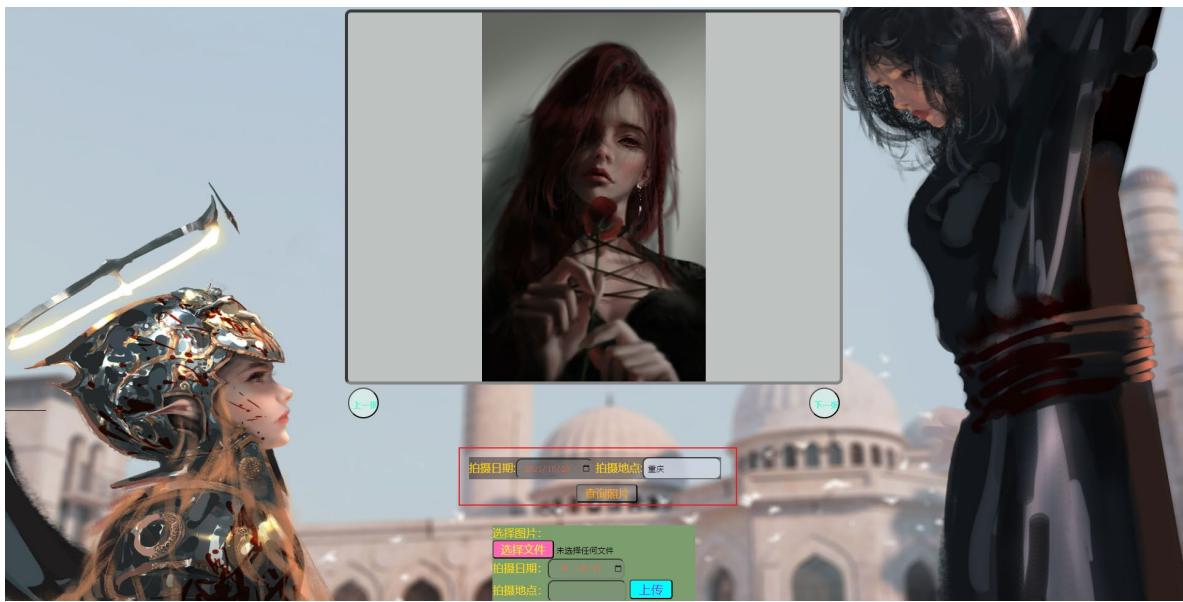


(4)

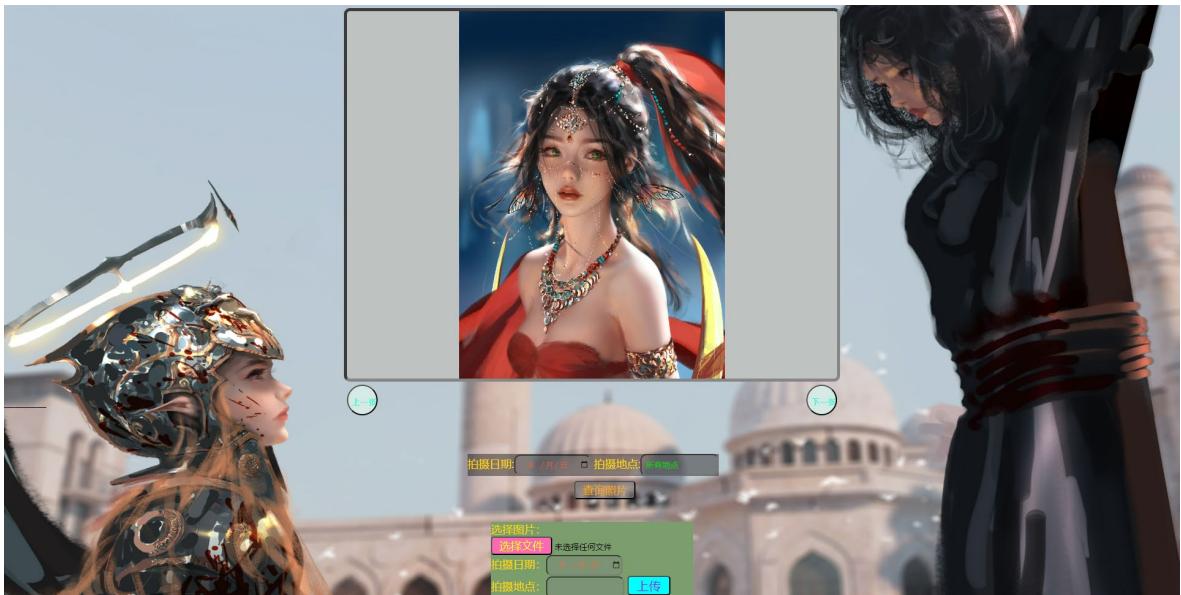


得到结果符合已存储图片情况，正确无误。

(9) 按日期和地点进行查询；



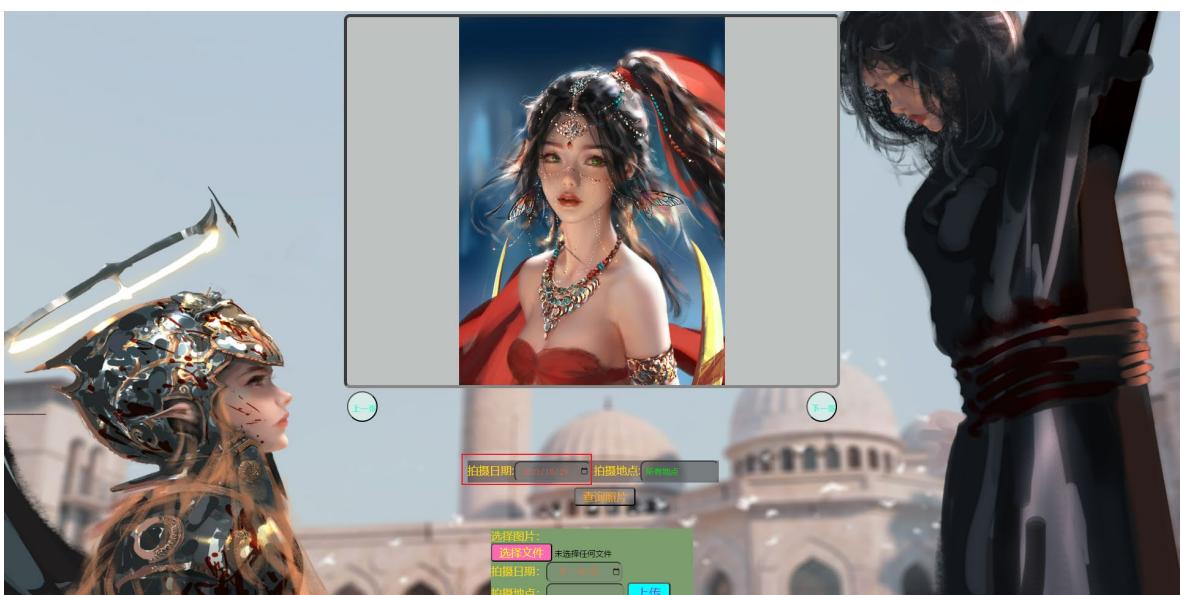
得到结果：



得到结果符合已存储图片情况，正确无误。

(10) 异常处理

①查询的日期不存在图片



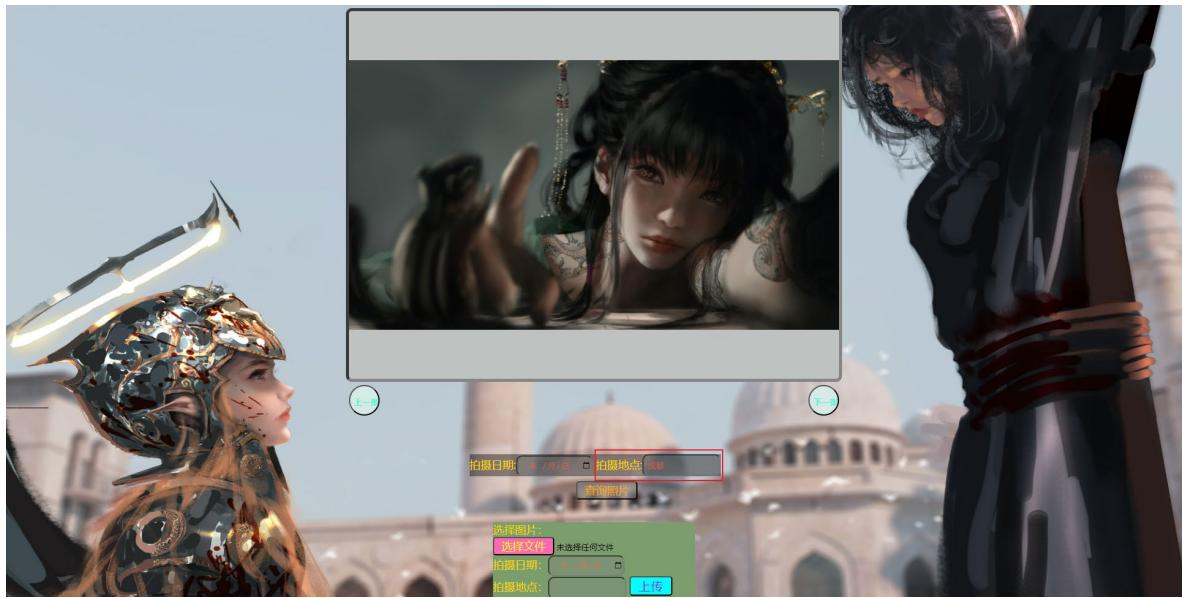
错误提示：

localhost:8080 显示

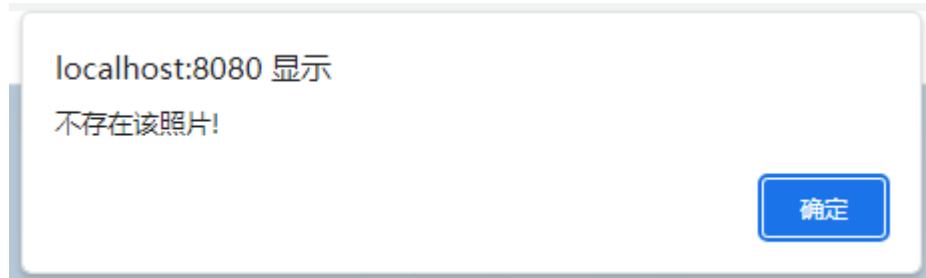
不存在该照片！

确定

②查询的地点不存在图片



会有错误提示:



2. 实验分析：

(1) 照片上传到服务器

我们首先利用 Tomcat 在本地建立了一个后端服务器，用于接收和存储用户上传的照片。

(2) 数据库设计

为了存储照片的信息，我们设计了一个数据库。这包括创建表以存储照片的路径、拍摄时间、拍摄地点等信息。使用 MySQL 数据库管理系统，我们执行了 SQL 语句来建立这个数据库结构。

(3) 查询功能

我们实现了查询功能，其中用户可以输入拍摄时间范围和地点信息。前端通过 HTTP 请求将查询条件发送给后端，后端使用 SQL 查询从数据库中检索符合条件的照片。

(4) 前端设计

我们灵活利用 HTML、CSS 和 JavaScript 等。设计了一个用户友好的前端界面，包括上传照片的表单、查询条件设置、图片浏览等。

(5) 响应式设计

照片浏览器页面经过响应式设计，能够适应不同设备和屏幕尺寸，在不同的显示器上均能正常显示，确保了用户体验的一致性。

(6) 总结和展望

通过完成这个实验，我们可以实现一个功能丰富的照片管理系统，用户能够方便地上传、浏览和查询照片。这个系统可以改善照片的管理和检索，使用户能够更轻松地找到他们需要的照片。

这个实验为我提供了宝贵的经验，包括 Web 开发、前端和后端协作、数据库设计和查询、文件上传安全、JavaScript 编程等技能，使我能够更好地理解和应用 Web 开发和数据库管理的原理。

未来，可以尝试改进自动获取照片信息的功能，以更准确地提取拍摄时间和地点，并处理各种图像格式，还可以进一步改进照片浏览器的 UI 页面，添加更多的创新设计元素，以进一步提高用户体验。此外，也可以考虑将该页面与其他功能集成，例如允许用户为照片添加标签或分类，以便更容易组织和查找特定主题的照片。

实验报告打印格式说明

1. 标题：三号加粗黑体
2. 开课实验室：5号加粗宋体
3. 表中内容：
 - (1) 标题：5号黑体
 - (2) 正文：5号宋体
4. 纸张：16开(20cm×26.5cm)
5. 版芯

上距：2cm

下距：2cm

左距：2.8cm

右距：2.8cm

说明：1、“年级专业班”可填写为“00电子1班”，表示2000级电子工程专业第1班。

2、实验成绩可按五级记分制（即优、良、中、及格、不及格），或者百分制记载，若需要将实验成绩加入对应课程总成绩的，则五级记分应转换为百分制。