实验 08：实现视频播放

一、实验目的

1、掌握 Vue 的基本概念；

2、掌握 Vue 的语法；

3、掌握 Vue 的指令。

二、实验学时

2 学时

三、实验类型

综合性

四、实验需求

1、硬件

每人配备计算机 1 台，建议优先使用个人计算机开展实验。

2、软件

安装 Visual Studio Code，以及 Edge 浏览器。

3、网络

本地主机能够访问互联网和实验中心网络。

4、工具

无。

五、实验任务

1、完成代办事项示例；

2、完成视频播放实战。

六、实验内容及步骤

1、完成代码事项示例

步骤 1：创建 todo-management 文件夹，并使用 Visual Studio Code 打开文件夹，如图 1 所示。

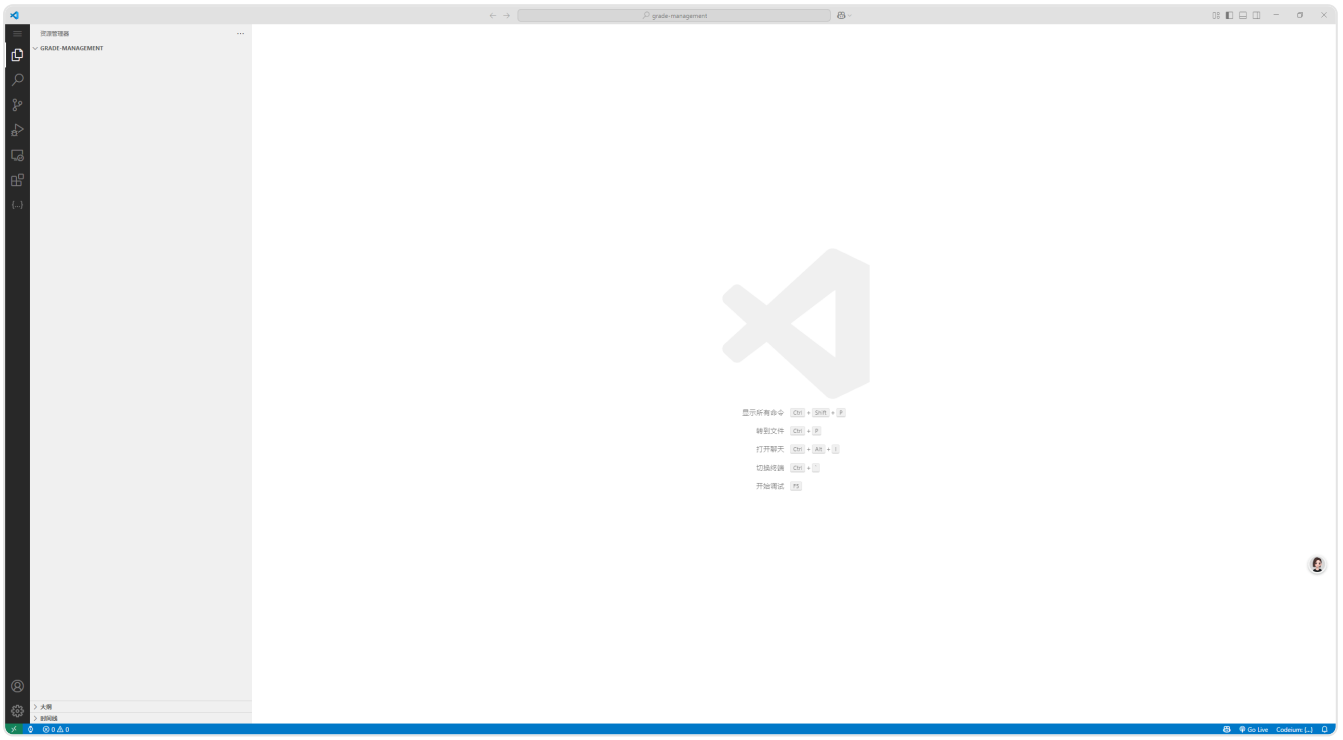

图1 使用Visual Stidio Code打开文件夹

步骤 2：在 Visual Studio Code 中打开终端，并执行以下命令创建 Vue 项目，如图 2 所示。

```Bash
1 npm create vue@latest
```

```
PS C:\Users\leile\Desktop\第9章\todo-management> npm create vue@latest

> npx
> create-vue

┌  Vue.js - The Progressive JavaScript Framework
│
◇  请输入项目名称：
│  todo-list
│
◇  请选择要包含的功能： (↑/↓ 切换，空格选择，a 全选，回车确认)
│  none
正在初始化项目 C:\Users\leile\Desktop\第9章\todo-management\todo-list...
└  项目初始化完成，可执行以下命令：

   cd todo-list
   npm install
   npm run dev

| 可选：使用以下命令在项目目录中初始化 Git：

   git init && git add -A && git commit -m "initial commit"
```

图2 创建Vue项目

步骤 3：执行以下命令安装项目依赖，执行结果如图 3 所示。

Bash

```
1 cd todo-list
2 npm install
```

```
PS C:\Users\leile\Desktop\第9章\todo-management> cd .\todo-list\
PS C:\Users\leile\Desktop\第9章\todo-management\todo-list> npm install

added 143 packages in 7s

42 packages are looking for funding
  run `npm fund` for details
```

图3 安装依赖

步骤 4：执行以下命令运行 Vue 项目，执行结果如图 4 所示。

Bash

```
1 npm run dev
```

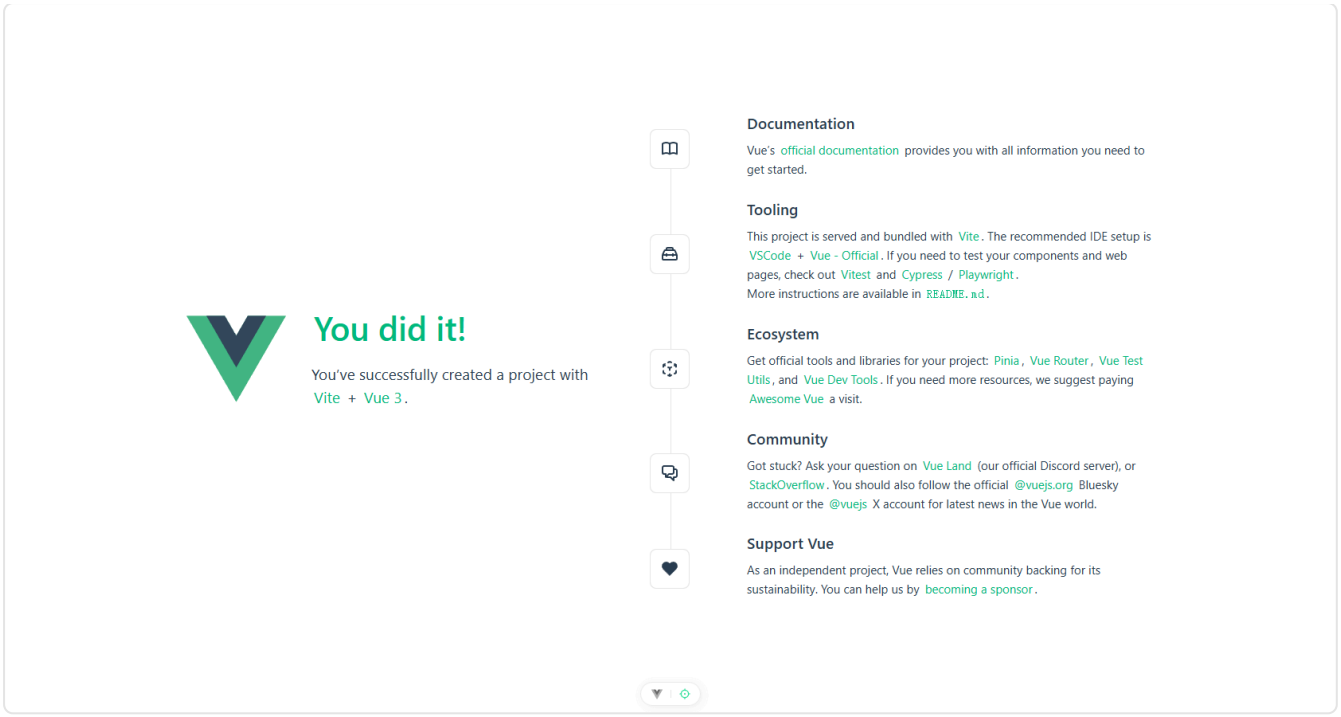

图4 Vue运行结果

步骤 5：修改 App.vue 代码，实现代办事项功能。

XML/HTML

```
<template>
  <div id="app">
    <h1>我的待办清单 📝</h1>

    <div class="input-section">
      <input type="text" v-model="newTask" @keyup.enter="addTask" placeholder="输入任务后按回车添加">
      <button @click="addTask">添加</button>
    </div>

    <transition-group name="fade" tag="ul">
      <li v-for="task in tasks" :key="task.id" :class="{ completed: task.completed }">
        <input type="checkbox" v-model="task.completed">
        <span>{{ task.content }}</span>
        <button @click="removeTask(task.id)">×</button>
      </li>
    </transition-group>

    <div class="stats">
      <p>总任务: {{ totalTasks }}</p>
      <p>未完成: {{ remainingTasks }}</p>
    </div>
  </div>
</template>

<script setup>
import { ref, computed, watch, onMounted } from 'vue'

const newTask = ref('')
const tasks = ref([])

// 计算属性
const totalTasks = computed(() => tasks.value.length)
const remainingTasks = computed(() => tasks.value.filter(t => !t.completed).length)

// 方法
const addTask = () => {
  if (newTask.value.trim()) {
    tasks.value.push({
      id: Date.now(),
```

```
      content: newTask.value.trim(),
      completed: false
    })
    newTask.value = ''
  }
}

const removeTask = (id) => {
  tasks.value = tasks.value.filter(t => t.id !== id)
}

// 生命周期钩子
onMounted(() => {
  const saved = localStorage.getItem('tasks')
  if (saved) tasks.value = JSON.parse(saved)
})

// 数据监听
watch(
  tasks,
  (newVal) => {
    localStorage.setItem('tasks', JSON.stringify(newVal))
  },
  { deep: true }
)
</script>

<style scoped>
.fade-enter-active,
.fade-leave-active {
  transition: all 0.5s;
}

.fade-enter,
.fade-leave-to {
  opacity: 0;
  transform: translateX(30px);
}

.completed {
  color: #888;
  text-decoration: line-through;
}

```

```
84 li {
85   transition: all 0.3s;
86   padding: 8px;
87   border-bottom: 1px solid #eee;
88 }
89 </style>
```

步骤 7：通过浏览器查看功能，如图 5 所示。

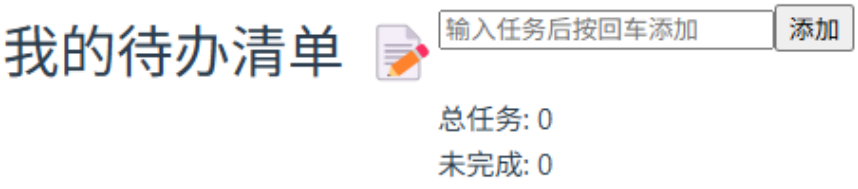

图5 查看代办清单功能

2、实现视频播放实战

实现一个功能完整、交互友好、支持多场景的视频播放页，支持播放、暂停、进度条、音量控制和全屏等功能，其详细要求如下：

- 基本功能
  - 使用 Vue 创建一个视频播放器页。
  - 提供播放和暂停按钮，点击按钮可以控制视频的播放和暂停状态。
  - 显示当前视频的播放进度，并提供一个进度条，允许用户拖动进度条来调整视频的播放位置。
- 进度条功能
  - 实现一个动态更新的进度条，显示视频的播放进度。
  - 当用户拖动进度条时，视频应该跳转到对应的时间点。
- 音量控制
  - 提供一个音量控制滑块，通过滑块调整视频的音量。
  - 音量滑块的值应该实时更新到视频元素的音量属性。
- 全屏功能
  - 提供一个全屏按钮，点击后视频进入全屏模式。
  - 在全屏模式下，用户仍然可以控制视频的播放、暂停和音量。
- 播放速度调节

    - 提供一个下拉菜单，允许用户选择不同的播放速度（如 0.5x、1x、1.5x、2x）。
    - 当用户选择不同的速度时，视频的播放速度应该实时调整。
- 事件处理
    - 使用 Vue 的事件处理机制（如 `v-on` 或 `@`）来绑定播放、暂停、进度条拖动、音量调整等事件。
    - 在视频加载完成和播放结束时，分别触发相应的事件并显示提示信息。
- 样式要求
    - 使用 CSS 对视频播放器进行样式美化，确保界面简洁美观。
    - 播放控制按钮和进度条需要有悬停和点击效果。
- 响应式设计
    - 确保视频播放器在不同屏幕尺寸下能够正常显示和操作。
    - 在移动设备上，视频播放器应该能够自适应屏幕宽度。

七、实验考核

本实验考核采用【实验随堂查】方式开展。

每个实验完成后，在实验课上通过现场演示的方式向实验指导教师进行汇报，并完成现场问答交流。

每个实验考核满分 100 分，其中实验成果汇报 60 分，现场提问交流 40 分。

实验考核流程：

（1）学生演示汇报实验内容的完成情况，实验指导老师现场打分。

（2）指导老师结合实验内容进行提问，每位学生提问 2-3 个问题，根据回答的情况现场打分。

（3）实验考核结束后，进行公布成绩。