**数据库、网络和网页**  
**期中作业 (2024年10月期)**  
**学生说明**  
**事件管理器**  
**简介**  
您的任务是创建一个可部署的事件管理器，用于组织事件。例如，一位瑜伽教练可以安排瑜伽课程，或者一个艺术画廊可以举办关于艺术作品的讲座。  
该事件管理器将用于单个个人或组织，并可以部署到 Web 服务器上。  
该系统应该有两个独立的页面：组织者页面和参与者页面。组织者页面用于设置和发布事件，参与者页面用于预订已发布的事件。  
请仔细阅读基础要求以获取完整的细节。

**技术规范**  
所有提交内容必须使用作业中提供的模板项目。服务器端功能必须基于 Express.js。数据层必须基于 SQLite。客户端页面必须通过服务器端渲染使用嵌入式 JavaScript 模板进行呈现。因此，以下工具是前提要求：

| **工具** | **描述** | **URL** |
| --- | --- | --- |
| EJS | 嵌入式 JavaScript 模板工具，支持动态渲染 HTML。 | [npm: ejs](https://www.npmjs.com/package/ejs) |
| SQLite3 | 在 Express 应用中查询数据层。 | [npm: sqlite3](https://www.npmjs.com/package/sqlite3) |
| ExpressJS | 用于简化构建 Web 服务器的框架。 | [Express](https://expressjs.com/) |
| NodeJs | 用于构建 Web 服务器的 JavaScript 引擎。 | [Node.js](https://nodejs.org/en/) |
| SQLite | 便携式、单文件的 SQL 数据库。 | [SQLite](https://www.sqlite.org/index.html) |

您可以使用 NPM 插件和前端库，前提是它们有正当理由且不根本改变上述规范。一些示例包如下：

| **工具** | **描述** | **URL** |
| --- | --- | --- |
| Tailwind | 流行的 CSS 库。需要一些较为复杂的构建工具。 | [Tailwind CSS](https://tailwindcss.com/) |
| Bootstrap | 较老但仍流行的 CSS 库，易于直接使用。 | [Bootstrap](https://getbootstrap.com/) |
| Axios | 一个用于发送请求的库，支持客户端和服务器端。 | [Axios](https://www.npmjs.com/package/axios) |
| Assert | Node 内置模块，用于测试和基本数据验证。 | Assert |
| Joi | 用于改进数据验证的有用库。 | [Joi](https://www.npmjs.com/package/joi) |
| Express-Validator | 数据验证的更可扩展方法。 | Express Validator |
| Express Sessions | 用于存储用户登录状态的持久和安全存储。 | [Express Sessions](https://www.npmjs.com/package/express-session) |
| Date-Fns | 日期/时间格式化工具。 | [Date-Fns](https://date-fns.org/) |

**注意事项**

* 所有源代码必须是可读的。
* 不要使用像 Webpack 这样的打包工具。
* 您可以使用像 Tailwind 这样的 CSS 预处理器，前提是提交时提供已构建的 CSS 文件。
* 我们将执行 npm install、npm run build-db 和 npm run start 来运行您的项目。请确保您的项目仅依赖于这些命令运行，不需要进一步的编译或安装依赖。
* 请确保您的 Node 版本至少为 16.x，NPM 版本至少为 8.x。

**数据要求**

* 页面必须动态从数据库获取用户数据并填充。可以通过模板（如本课程所教）或其他方式（如 JavaScript 请求和 DOM 操作）来完成。
* 依赖硬编码客户端用户数据的功能将不被考虑。
* 所有用户数据必须持久化存储在 SQLite 数据库中。

**代码风格要求**

* 前端代码包含视图文件夹中的模板文件（.ejs），并可选地包含公共文件夹中的资产和 .js 文件。
* 后端代码包含 index.js 文件，路由以中间件的形式实现于路由文件夹中。
* 每个路由前应有注释，描述其目的、输入和输出。
* 路由应合理组织，适当使用 GET 和 POST 方法。
* 每个数据库交互应有注释，描述其目的、输入和输出。
* 代码布局清晰，缩进一致。
* 函数和变量应有意义的名称，并遵循一致的命名风格。
* 所有变量声明时应仔细考虑作用域。

**模板代码**  
所有提交必须使用提供的模板项目。请查看模板代码中的 README.md 文件，了解如何使用它。

**基础要求：**

**组织者首页**  
这是组织者可以创建、审阅和编辑事件的页面。该页面的最基本要求如下：

* 显示一个标题，指示这是组织者首页。
* 应通过一个与参与者首页不同的 URL 访问。
* 显示事件管理器的名称和描述，例如“Stretch Yoga”和“适合所有年龄段和能力的瑜伽课程”。
* 应有一个指向站点设置页面的链接。
* 应有一个“创建新事件”按钮。
  + 按下按钮时，创建一个新的草稿事件，并重定向到其编辑页面。
* 应显示一个动态填充的已发布事件列表。
  + 列表应显示有关事件的有用信息，包括标题、日期、创建和发布的时间以及每种类型的票的数量。
  + 每个事件应显示一个分享链接，指向相关的参与者首页。
  + 每个事件应有一个删除按钮。按下时：
    - 从数据库中删除该事件。
    - 重新加载页面，显示更新后的信息。
* 应显示一个动态填充的草稿事件列表。
  + 列表应显示有关事件的有用信息，包括标题、日期、创建和发布的时间以及每种类型的票的数量。
  + 每个事件应有一个指向其编辑页面的链接。
  + 每个事件应有一个发布按钮。按下时：
    - 将事件的状态从草稿更新为已发布。
    - 记录发布时间。
    - 重新加载页面，显示更新后的信息。
  + 每个事件应有一个删除按钮。按下时：
    - 从数据库中删除该事件。
    - 重新加载页面，显示更新后的信息。

**站点设置页面**  
这是组织者可以更改站点名称和描述的页面。该页面的最基本要求如下：

* 显示一个标题，指示这是站点设置页面。
* 包含一个文本输入表单，用于名称和描述。
  + 表单应动态填充当前站点设置。
  + 表单应有一个提交按钮，用于更新设置并重定向到组织者首页。
  + 提交前应进行表单验证，确保所有字段已填写。
* 包含一个指向组织者首页的返回按钮。

**组织者编辑事件页面**  
这是组织者创建或修改事件的页面。该页面的最基本要求如下：

* 显示一个标题，指示这是组织者编辑事件页面。
* 显示事件的创建日期。
* 包含一个表单，包含以下输入字段和控件：
  + 事件标题
  + 事件描述
  + 正常票数量和价格
  + 优惠票数量和价格
  + 提交更改按钮
* 编辑现有事件时，表单应填充当前事件数据。
* 提交更改时，事件的最后修改日期应更新。
* 包含一个指向组织者首页的返回按钮。

**参与者首页**  
这是用户可以查看并预订事件的前端页面。该页面的最基本要求如下：

* 显示一个标题，指示这是参与者首页。
* 应通过一个与组织者首页不同的 URL 访问。
* 显示站点名称和描述。
* 显示已发布的事件列表：
  + 每个事件应显示标题和日期。
  + 事件应按事件日期排序，最近的事件应出现在最上面。
  + 点击列表中的一个事件，应跳转到该事件的参与者页面。

**参与者事件页面**  
这是参与者可以查看事件详细信息并进行预订的页面。该页面的最基本要求如下：

* 显示一个标题，指示这是参与者事件页面
* 应根据 URL 显示一个特定事件
* 显示事件的详细信息，包括标题、描述、日期、票种和票价
* 用户应能够选择要购买的每种类型的票数
* 用户应输入自己的姓名
* 应有一个“预订”按钮，用于完成票务预订
* 参与者只能预订剩余票数以内的票
* 应有一个返回按钮，将用户重定向到参与者首页

**主首页**  
这是应用程序的默认首页，可以通过 http://localhost:3000/ 访问。首页应简单包含以下内容：

* 一个指向组织者首页的链接
* 一个指向参与者首页的链接

**扩展功能：**  
您还可以通过以下选项之一来增强项目。您可以完成多个扩展，但我们仅会考虑评论中指定的第一个扩展。

**添加额外功能：**

* 向应用程序的组织者端和参与者端添加额外功能。例如：
  + 添加设置不同票种的功能
  + 向组织者和参与者显示剩余票数
  + 添加一个页面，允许组织者查看所有已完成的预订

**组织者页面和路由的密码访问：**

* 使用密码身份验证来保护组织者页面，并编写中间件防止未授权访问组织者端点
* 您需要创建一个组织者登录页面，通过简单的服务器端密码验证（例如在环境变量中存储）来验证组织者身份
* 您还需要使用如 express-session 的包来创建安全的会话

**最佳实践：**  
将代码库朝着生产级别发展，遵循以下推荐的最佳实践：

* 实施来自 [Express 安全最佳实践](https://expressjs.com/en/advanced/best-practice-security.html) 的相关安全最佳实践
* 实施来自 [Express 性能最佳实践](https://expressjs.com/en/advanced/best-practice-performance.html) 的代码性能最佳实践
* 使用像 [Joi](https://joi.dev/) 或 [express-validator](https://express-validator.github.io) 等库进行数据净化
* 确保在您的项目中列出所有使用的额外库

**交付要求：**

1. **源代码**  
   请上传您的完整源代码（zip 格式）。

* 确保项目中不包含 node\_modules 目录
* 确保 package.json 文件包含所有额外安装的包
  + 提示：确保使用 npm install --save package\_name 安装它们
* 不要包括您的 SQLite 数据库文件（.db）。我们将通过运行 npm build-database 来构建数据库
* 提交一个 README.md 文件
  + 移除我们提供的现有内容，只保留关于如何设置应用程序的简单说明
  + 列出您使用的任何额外库

1. **报告（PDF 格式）**  
   请上传报告（PDF 格式，最多 1000 字）。

* 网站架构的高层次示意图，展示客户端与服务器之间的所有端点
  + 使用 Figma (<https://figma.com>) 或 Lucid Chart ([<https://lucid.app/>)) 等免费工具制作
* 数据模型的高层次示意图，绘制 ER 图，展示表和它们之间的关系，使用 “crow's feet” 符号表示
  + 使用 Figma 或 Lucid Chart 制作
* 扩展功能描述
  + 指定您实施的扩展功能
  + 讨论您是如何实现它的
  + 强调您希望我们特别关注的方面
  + 粘贴相关的代码片段，并解释如何实现扩展。请标注文件名和行号（如果您没有提供这些，可能会丧失分数）

1. **视频录屏**  
   请上传一个演示项目工作的录屏视频。

* 视频时长不得超过 2.5 分钟（我们不会观看超过 2.5 分钟的视频）。
* 视频应展示所有功能
* 推荐使用如 OBS 等软件以 mp4 格式录制视频
* 这不是一个评分要求，而是为了防止我们无法安装和运行您的代码时使用的备份。您必须提交一个视频，确保我们能够评分您的代码

1. **代码（PDF 格式）**  
   请上传您的代码（以文本格式，不是截图），并以 PDF 文件提交。

* 清楚地标记代码的开始和结束部分，明确指出哪些代码是您自己编写的，而没有任何帮助。这将帮助我们进行必要的抄袭检查。

您如果没有按要求提交代码，整个项目将得零分。

**成功的提示：**

* 仔细阅读上述要求，并确保完全按要求实现它们。
* 为代码添加良好的注释。注释每个路由端点和每个函数，描述它们的目的，列出输入和输出。
* 提交前，请测试所有应用程序的功能。
* 提交前，请测试您的部署过程。删除 node\_modules 目录，然后运行：npm install、npm run build-db（或 npm run build-db-win）和 npm run start。
* 特别注意确保 db\_schema.sql 脚本能创建您的数据库，不会出现任何错误或问题。

**评估标准：**  
您必须按要求提供 PDF 格式的源代码。如果未提交，将导致整个项目得零分。

1. **功能性（40分）**

* 主首页
* 组织者首页功能
* 组织者设置页面
* 组织者编辑事件
* 参与者首页
* 参与者事件页面

1. **实现质量（30分）**

* 前端可用性和样式
* 后端代码质量
* 数据库设计
* README 和注释

1. **文档和扩展（30分）**

* 架构图
* 数据库图
* 扩展功能的雄心
* 扩展功能实现
* 扩展功能的实现质量
* 扩展功能文档