**FIT2095 Assignment 2 Documents**

Group Name: Lab04 -Team16

Students: #1 Malcolm Paltiraja (32578393)

#2 Cherylyn Yong Jo Yee (32714491)

**Routing table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Name** | **Method** | **Path Name** | **Description** |
| 1 | Home Page | GET | / | Renders the index.html file. Contains five buttons for each student to link to the other webpages. |
| 2 | Add Category | GET | /category/32578393/add-category | Responds with the add-category page. Client can fill the details through form. |
| 3 | Add Category | POST | /category/32578393/add-category | Extracts the input from request’s body, auto generated random Id and save them in an array. This endpoint will redirect the client to list category page. |
| 4 | List all categories | GET | /category/32578393/list-category | Renders list-category.html. It will show the list of category attributes in a table. There is a button in the last column of the table to redirect the client to show category details. |
| 5 | List categories by keyword | GET | /category/32578393/list-keyword/:keyword | The keyword will be extracted from query string. Then, it renders the list category page which the categories contain keyword in description. The category attributes will be shown in table. |
| 6 | Show Events Details | GET | /category/32578393/view-event/:eventId | Renders view-event.html. All details of event will be shown to the client. |
| 7 | Delete Category by ID | GET | /category/32578393/delete-category | Responds with the delete-category page. Client can fill the input with the category that they wish to delete. |
| 8 | Delete Category by ID | POST | /category/32578393/delete-category | Extracts the input from request’s body, filter out the category that wish to be deleted from array. This endpoint will redirect the client to list category page. |
| 9 | Add Event | GET | /cherylyn/event/add | Responds with the add-event page. Client can fill the details through form. |
| 10 | Add Event | POST | /cherylyn/event/add | Extracts the input from request’s body, auto generated random Id and save them in an array. This endpoint will redirect the client to list category page. |
| 11 | List all events | GET | /cherylyn/event/listall | Renders list-event.html. It will show the list of event attributes in a table. There is a button in the last column of the table to redirect the client to show event details. |
| 12 | List Sold-out Events | GET | /cherylyn/event/soldout | Renders sold-out-event.html. It will show all the sold-out events (ticketAvailable = 0) in a table. There is a button in the last column of the table to redirect the client to show event details. |
| 13 | Category Detail Page | GET | /cherylyn/category/:categoryID | Renders category-detail.html. All details of details will be shown to the client. Underneath category attributes, all events of that category will be listed in a table. |
| 14 | Delete Event by Id | GET | /cherylyn/event/remove/ | Extracts idDelete from the query string, filter out the event that wish to be deleted from array. This endpoint will redirect the client to list event page. |
| 15 | Create Category | POST | /api/v1/category/32578393/add | To add a new category. The data is sent through the body of the HTTP request in JSON format. The endpoint is responded with a JSON object containing the category ID of the newly added category. |
| 16 | List All Categories | GET | /api/v1/category/32578393/list | The endpoint return list of all categories and details for events in JSON format. |
| 17 | Delete category by ID | DELETE | /api/v1/category/32578393/delete | The endpoint deletes a category by its ID and all the events that are listed in the ‘eventslist’. The ID is sent through the body of the request in a JSON object and the endpoint will return a JSON object containing number of deleted documents. |
| 18 | Update category name and description by ID | PATCH | /api/v1/category/32578393/update | Update category name and description by ID. New ID, name and description are sent as a JSON object through the request body and the endpoint will return an object with a message to confirm the update or if the ID is not found. |
| 19 | Create Event | POST | /cherylyn/api/v1/createEvent | To add a new event. The data is sent through the body of the HTTP request in JSON format. The endpoint is responded with a JSON object containing the event ID of new event. |
| 20 | List All Events | GET | /cherylyn/api/v1/getAll | The endpoint return list of all events and details for their categories in JSON format. |
| 21 | Delete event by ID | DELETE | /cherylyn/api/v1/deleteById | The endpoint deletes a event by its ID. The ID is removed from the ‘eventList’ array in all categories that are listed in the ‘categoryList’. The ID is sent through the body of the request in a JSON object and the endpoint will return a JSON object containing number of deleted documents. |
| 22 | Update event name and capacity by ID | PUT | /cherylyn/api/v1/updateById | Update event name and capacity by ID. New ID, name and capacity are sent as a JSON object through the request body and the endpoint will return an object with a message to confirm the update or if the ID is not found. |

**Steps to Run Application**

1. Go to home page using *localhost:8080/.*
2. Click the ‘add category’ button to add category. Alternatively, you can also enter the URL *localhost:8080/category/32578393/add-category* to go to add-category page.
3. Fill in all the details for the category and click submit. The category will be recorded and listed in table.
4. If you wish to see the details of specific category, you can click on the ‘view’ button at the end of the table. Category details and events recorded in that category will be displayed. You can also check the details of category by typing the URL *localhost:8080/cherylyn/category/:categoryID* which :categoryID is the category ID that you interested.
5. Go to home page again and click on ‘Add Event’ button to add an event. Alternatively, you can also enter the URL *localhost:8080/cherylyn/event/add* to go to add-event page.
6. Fill in all the details for the event and click submit. The event is recorded and listed in table.
7. If you wish to see the details of specific event, you can click on the ‘view’ button at the end of the table. All the details of event will be presented to you. You can also check the details of event by typing the URL *localhost:8080/category/32578393/view-event/:eventId* which :eventId is the event ID that you interested.
8. You can search for specific category using keywords. Just type *localhost:8080/category/32578393/list-keyword/:keyword* where :keyword is the desired keyword you want to find from description of categories.
9. If you want to see if the event that you interested sold out or not, you can go to sold-out page. There are two ways for you to do so, first, click the ‘home’ button and it will redirect you to the home page. Then, you can click on the ‘List Sold Out Events’ button, it will redirect you to the sold-out page and you will be able to determine if the event is sold-out. If you think the first method is inconvenient, you can enter the URL *localhost:8080*/*cherylyn/event/soldout.* It will straight redirect you to the sold-out page.
10. You can delete a category. First, click on the ‘Delete Category by ID’ button and it will redirect you to delete-category page. Enter the category you wish to delete and press ‘Submit’. You can see that the category is deleted. The alternative way is entering the URL *localhost:8080/category/32578393/delete-category.*
11. You can delete an event as well. Enter the URL *localhost:8080/cherylyn/event/remove/?id=xxx-xxxx*. The xxx-xxxx is the event ID that you wish to be deleted. After entering the link, you will see that the event is deleted.
12. There are the other ways which is using postman to check the url.