Assignment 1

Things you will learn 🎖️

In this assignment, you will learn the following:

- JavaScript, HTML and CSS
- Develop a backend server using Node.js and Express.js
- Serve HTTP requests: GET, PUT, POST, and DELETE
- Respond with HTML files
- Push to GitLab Repository
- Work in teams

Assignment Theme 🎬

Monash has decided to start sports tournaments across its faculties and asked the IT department to provide all the technical support. There are different types of sports and different groups of staff. The IT department released the need to have multiple events grouped into categories. Due to not having a tool that manages events in categories, the IT department hired a software company called MonashApps to develop an Events Management Application (EMA).

Now, suppose you are working as a developer at MonashApps company and a member of the team that builds the **EMA** app from scratch based on specifications provided by the Monash IT department. This app is responsible for creating, managing and listing events in categories. The app is requested to be a web application that all devices with web browsing capability can access.

Assignment 1 Specifications

Assignment 1 is due on Friday of Week 5, August 25, at 11:55 pm.

Assignment 1 interview will be conducted in labs week 6

As mentioned in the Assignment Theme section, this assignment is about developing an Events Management Application (EMA) implemented as a web app. The development will occur in three stages (assignments), and now, let's focus on the first stage (Assignment 1), where we build the backend server to support EMA.

You will develop the EMA backend server in this assignment. The app serves the clients' HTTP requests and responds with HTML files.

Temporary Data Storage
You must use an array of objects (see A1 support) to hold your data temporarily.

The EMA should be designed to store **events** that are grouped into **categories.** Each event and category is defined by a set of attributes provided in Tables 1 and 2 below:

The column '**Required**' indicates whether the field's value is optional or mandatory. For example, the app should be able to save a new event without having a value for the '**Description**' field.

Event Category (Table 1)

Events (Table 2)

Teamwork

- produce together the overall app design/development plan
- produce together the team tasks
- agree on **who does which** set of individual tasks
- check in regularly with one another
- anything else about how you'd like them to work together [e.g. git commit authorship, branches, notes from meetings, Trello boards, etc.]

Individual work: you should produce the individual tasks yourself but should discuss ideas with your team

Team Tasks

1. **Develop a backend server** that uses Node.js (See Week 2 Lecture) and Express.js (see Week 3 Lecture) to serve clients' requests. The server must listen to a port specified by the team.
2. **Design the endpoints routing table.** In this table, you must list all the endpoints your project will listen to, their methods, and their description (see Week 3 Lecture/Routing Table). It's your task to name the pathnames, endpoints, and query string parameters.
3. **Represent Tables 1 & 2 in classes/objects** (see HERE for help).
4. If a **request with no pathname** arrives at the backend server (e.g. http://localhost:7823), the server must respond with the index.html file or app's home page (see Week 4 Lecture/Express and the static assets).
5. The team is responsible for **developing the index.html file**, which should contain the following:
    1. A Link to each individual-work requirement (listed below).
    2. A logo for the team served as a static asset. The logo is a small image that could contain team members' names, the team's icon, and/or slogan. You can use any drawing tool, such as MS PowerPoint, Google Slides, or Google Drawing.

Individual-Work Tasks

Each student in the team must pick one group of individual tasks to implement. However, cooperation between the team members is crucial for all the specs and features to work seamlessly.

> Teams with three members will be emailed the thrid group of tasks separately.

1. **Tasks Group 1 (student #1)**
   You MUST use your Monash ID as a segment in the pathname of all the endpoints of this

group of tasks. For example, if your Monash ID is 543210, the endpoint could be "http://localhost:8080/category/543210/add".

   1. **Add Category**: users use this page to add a new Category to the app. A page with a form containing all the category's attributes (see Table 1) and a button to submit the form data to the backend server are required. The endpoint must redirect the client to "List all categories". (Ref: see week 4 lecture material for more details)
   2. **List all categories**: in this page, you must list all the categories entered in a tabular format. For each row, a button should redirect the client to show category details (student #2-Task 4).
   3. **List categories by keyword**: in this endpoint, the user sends a keyword in a query string, and the response should be a page containing all the categories with the keyword in their description (case-insensitive search). The page should use a tabular format to list the categories. For each row, a button should redirect the client to show category details (student #2-Task 4).
   4. **Show Events Details:** Event Detail Page Display all event attributes:
      - Banner Image in the header
      - Date/time fields formatted, e.g. 6/28/2023, 6:30:00 PM
      - Calculated End DateTime: StartDateTime + DurationInMinutes
      - Formatted Duration in minutes, a few examples: "45 minutes", "2 hours 30 minutes", "2 hours", etc.
   5. **Delete** Category By ID: This endpoint deletes a category by its ID. The ID is sent to the backend server through the request's body (See HERE for help on how to slice an array in JS). The endpoint must redirect the client to "List all categories".

2. **Tasks Group 2 (student #2)**

   You MUST use your first name as a segment in the pathname of all the endpoints of this group of tasks. For example, if your first name is Alex, the endpoint could be http://localhost:8080/alex/event/add.

   1. **Add Event**: users use this page to add a new Event to the app. A page with a form containing all the Event's attributes (see Table 2) and a button to submit the form data to the backend server are required (Note: use existing category Id while creating the event to categorize it). (Ref: see week 4 lecture material for more details)
   2. **List all events**: in this page, you must list all the events entered in a tabular format. For each row, a button should redirect the client to show event details (student #1-Task 4).
   3. **List Sold-out events: this page lists all sold-out events in a tabular format similar to** the "List all events" page above. For each row, a button should redirect the client to show event details (student #1-Task 4).
   4. **Category Detail Page:** Display category attributes in any format of your choice or a user-friendly manner. Underneath category attributes, list all the events of the category in a tabular format. The id of the category is to pass as a URL parameter. For example *http://localhost:8080/alex/category/CME-3339*
   5. **Delete Event By ID:** This endpoint deletes an event by its ID. The ID is sent to the backend server as a variable in the URL's query string. The endpoint must redirect the client to the "list all events" page. *Example: http://localhost:8080/alex/event/remove?id=EGB-3703*

Hint: Don't forget to use EJS for dynamic pages such as list events and categories.

Source Code Quality

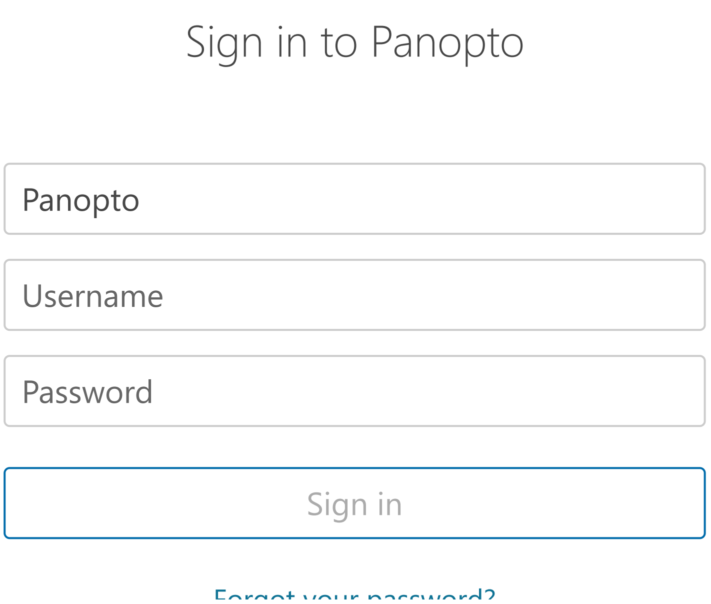
Two standards will be used to define and measure the quality of your source code.

1. Source Code documentation. You must use the JSDoc library to document your source code for this assignment. (See Assignment Control Panel-->Code Documentation and lab week 4 for help)
   You are required to provide documentation for your:
   1. classes
   2. global variables and constants
   3. methods
2. Naming Convention. The names of your files, classes, packages, modules, methods and variables must be descriptive and follow the Google naming style. (See Assignment Control Panel-->Code Style for more help)

UI Design Quality

Although Assignment 1 is mainly for backend development, some tasks required serving HTML pages. Therefore, you should provide a decent UI design for your pages and make them as friendly as possible. In this regard, you should use the Bootstrap library, which is delivered in weeks 3 and 4 (see labs weeks 3&4).

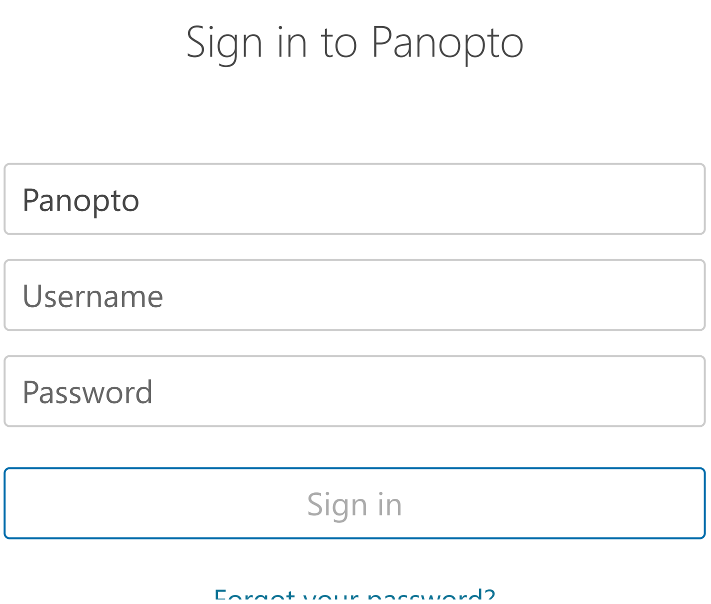
Expected Output



Sign in to Panopto

Panopto

Tasks Group 2 (Student#2)



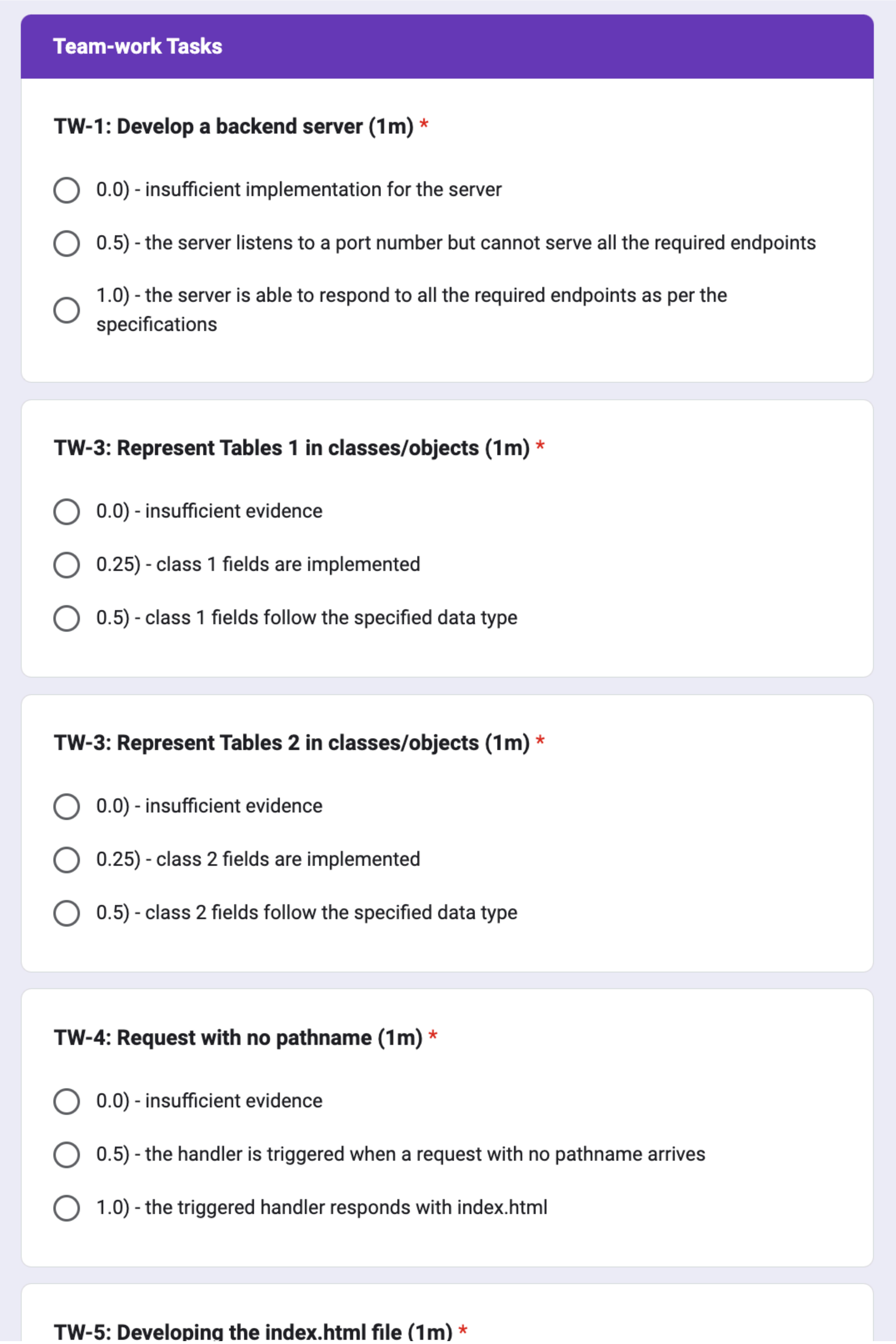
Sign in to Panopto

Panopto

Marking Rubric 🧑‍🏫

Summary

Team-work Tasks



Team-work Tasks

TW-1: Develop a backend server (1m) *

○ 0.0) - insufficient implementation for the server

○ 0.5) - the server listens to a port number but cannot serve all the required endpoints

○ 1.0) - the server is able to respond to all the required endpoints as per the specifications

TW-3: Represent Tables 1 in classes/objects (1m) *

○ 0.0) - insufficient evidence

○ 0.25) - class 1 fields are implemented

○ 0.5) - class 1 fields follow the specified data type

TW-3: Represent Tables 2 in classes/objects (1m) *

○ 0.0) - insufficient evidence

○ 0.25) - class 2 fields are implemented

○ 0.5) - class 2 fields follow the specified data type

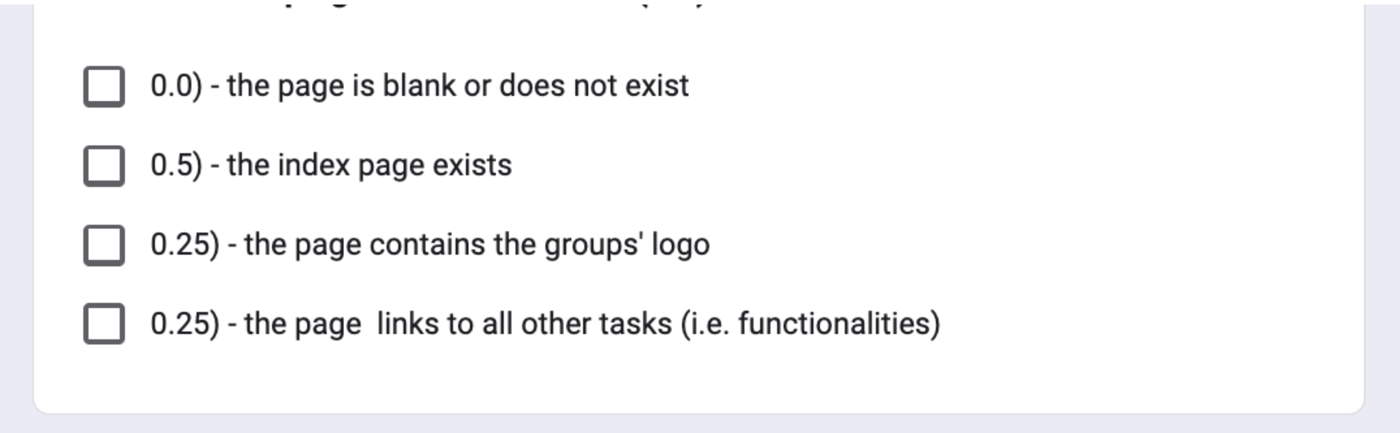
TW-4: Request with no pathname (1m) *

○ 0.0) - insufficient evidence

○ 0.5) - the handler is triggered when a request with no pathname arrives

○ 1.0) - the triggered handler responds with index.html

TW-5: Developing the index.html file (1m) *



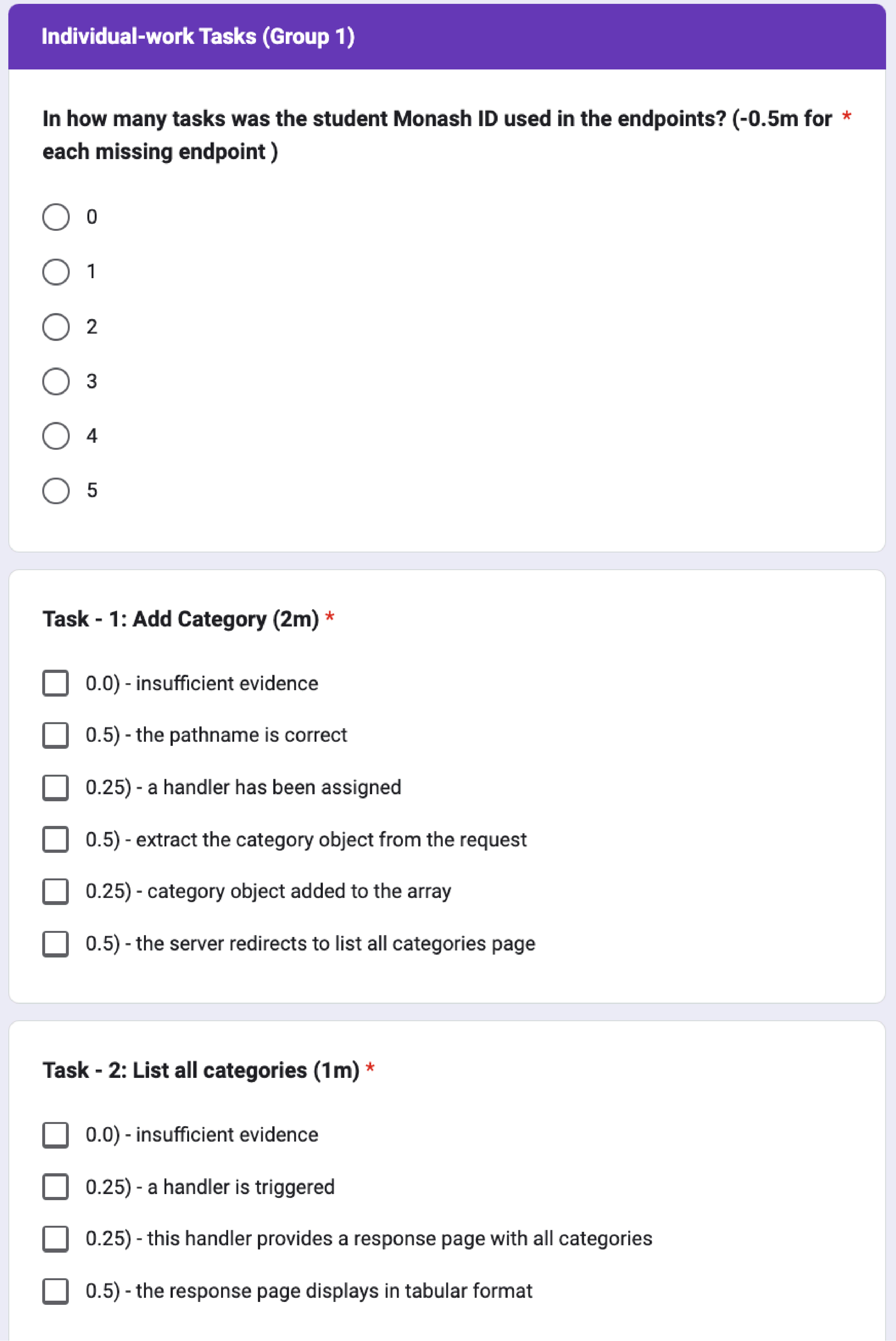
□ 0.0) - the page is blank or does not exist

□ 0.5) - the index page exists

□ 0.25) - the page contains the groups' logo

□ 0.25) - the page links to all other tasks (i.e. functionalities)

Individual Tasks (Student 1)



Individual-work Tasks (Group 1)

In how many tasks was the student Monash ID used in the endpoints? (-0.5m for each missing endpoint) *

○ 0

○ 1

○ 2

○ 3

○ 4

○ 5

Task - 1: Add Category (2m) *

□ 0.0) - insufficient evidence

□ 0.5) - the pathname is correct

□ 0.25) - a handler has been assigned

□ 0.5) - extract the category object from the request

□ 0.25) - category object added to the array

□ 0.5) - the server redirects to list all categories page

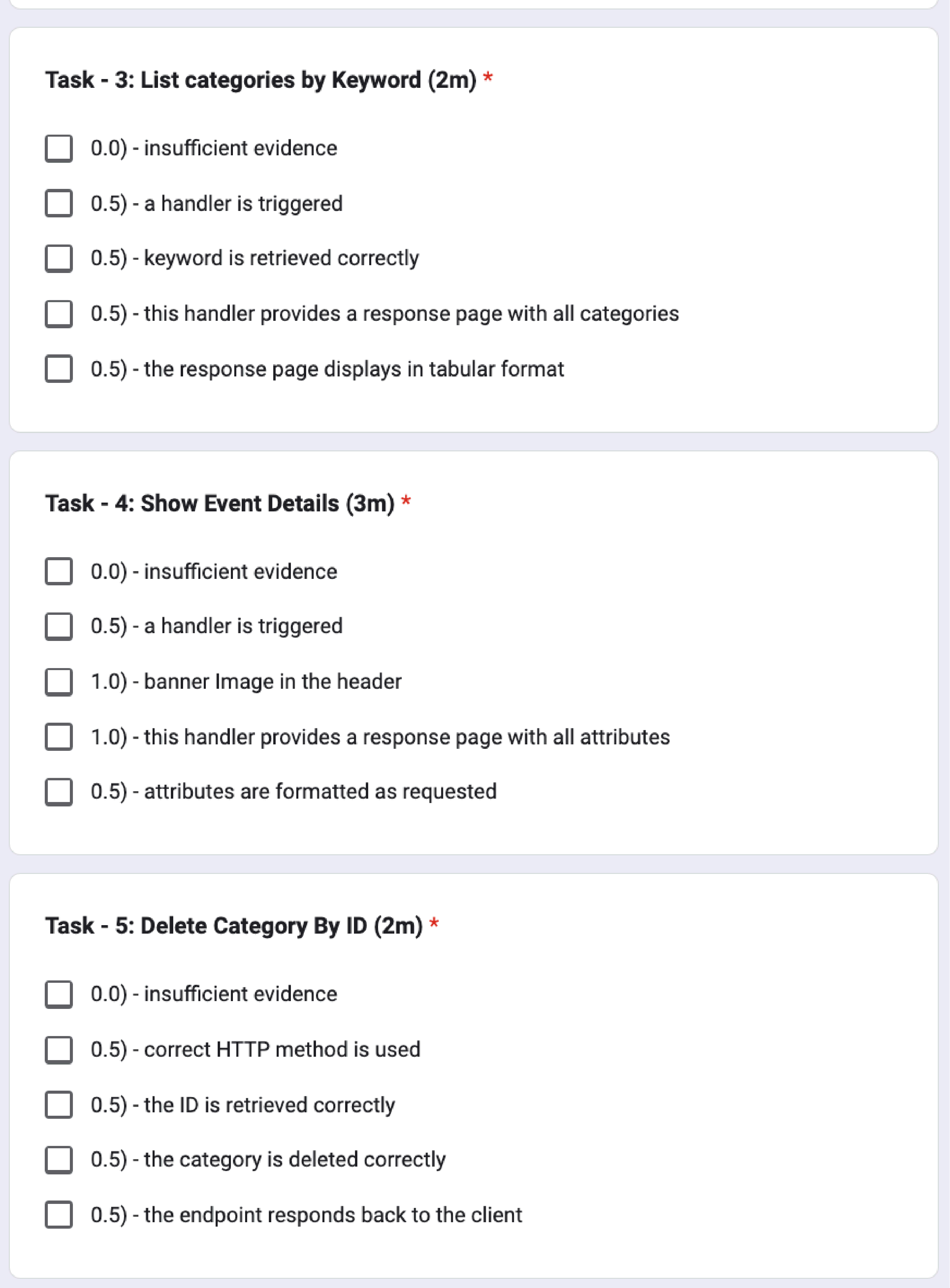
Task - 2: List all categories (1m) *

□ 0.0) - insufficient evidence

□ 0.25) - a handler is triggered

□ 0.25) - this handler provides a response page with all categories

□ 0.5) - the response page displays in tabular format



Task - 3: List categories by Keyword (2m) *

□ 0.0) - insufficient evidence

□ 0.5) - a handler is triggered

□ 0.5) - keyword is retrieved correctly

□ 0.5) - this handler provides a response page with all categories

□ 0.5) - the response page displays in tabular format

Task - 4: Show Event Details (3m) *

□ 0.0) - insufficient evidence

□ 0.5) - a handler is triggered

□ 1.0) - banner Image in the header

□ 1.0) - this handler provides a response page with all attributes

□ 0.5) - attributes are formatted as requested

Task - 5: Delete Category By ID (2m) *

□ 0.0) - insufficient evidence

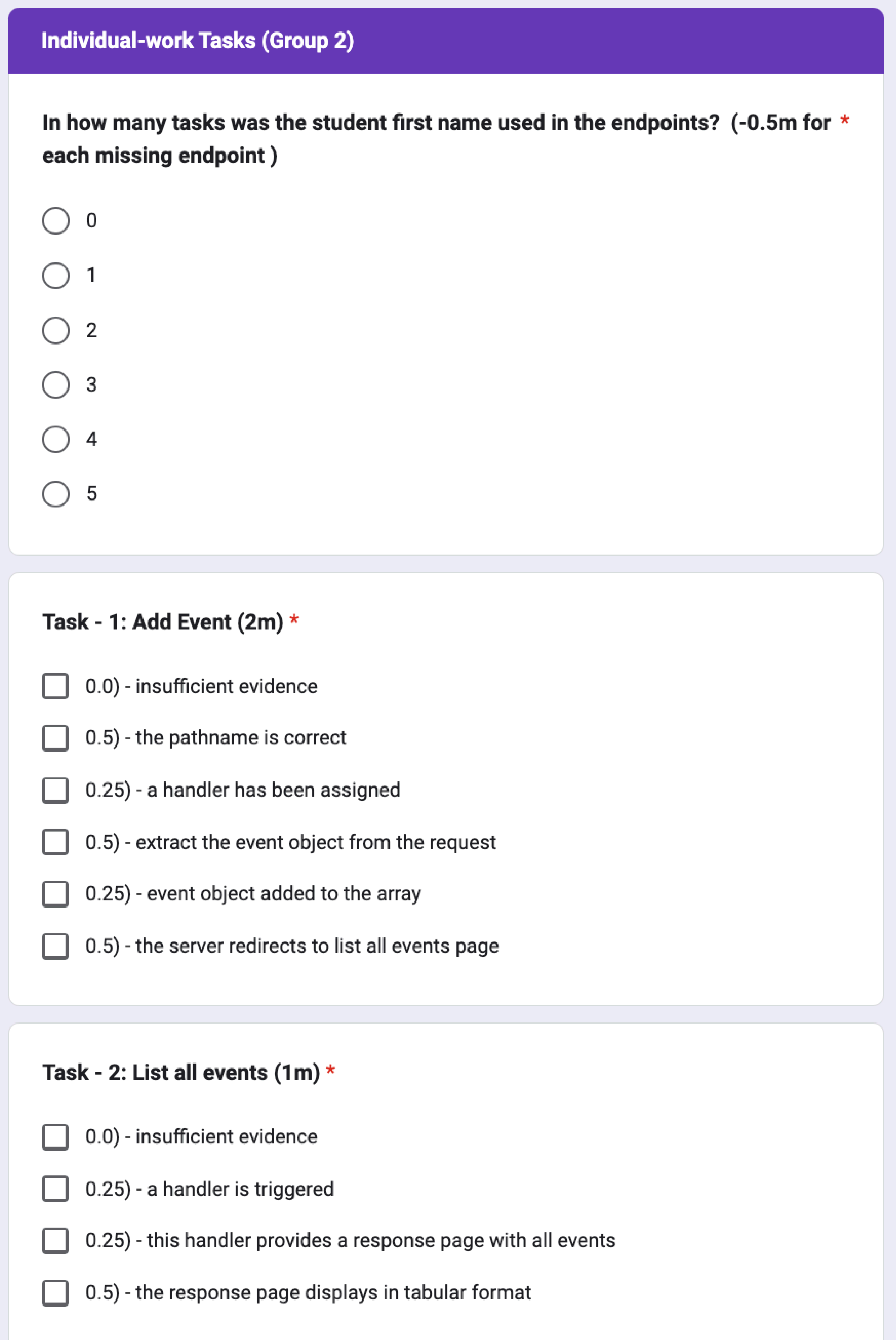
□ 0.5) - correct HTTP method is used

□ 0.5) - the ID is retrieved correctly

□ 0.5) - the category is deleted correctly

□ 0.5) - the endpoint responds back to the client

Individual Tasks (Student 2)



Individual-work Tasks (Group 2)

In how many tasks was the student first name used in the endpoints? (-0.5m for each missing endpoint) *

○ 0

○ 1

○ 2

○ 3

○ 4

○ 5

Task - 1: Add Event (2m) *

□ 0.0) - insufficient evidence

□ 0.5) - the pathname is correct

□ 0.25) - a handler has been assigned

□ 0.5) - extract the event object from the request

□ 0.25) - event object added to the array

□ 0.5) - the server redirects to list all events page

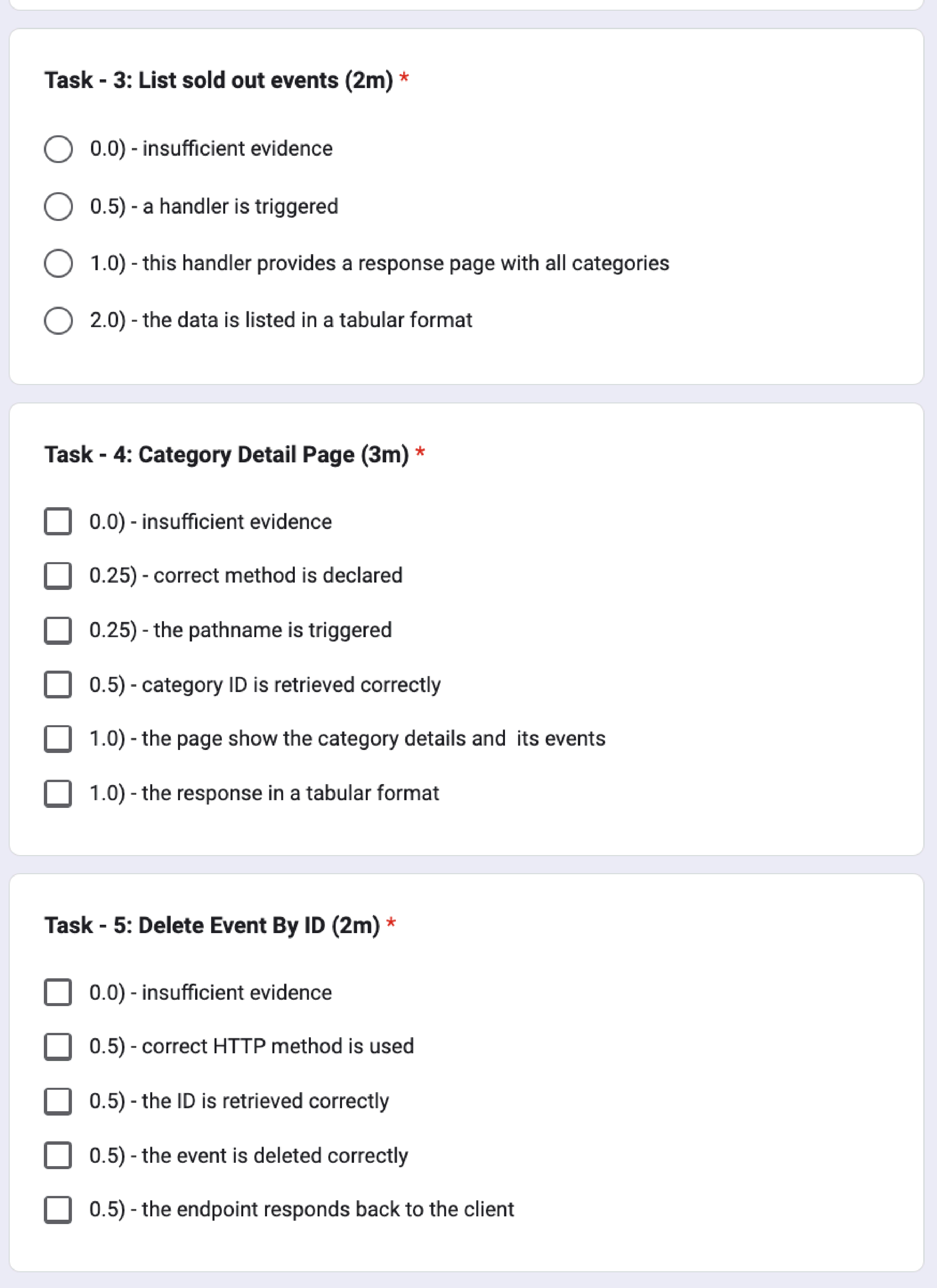
Task - 2: List all events (1m) *

□ 0.0) - insufficient evidence

□ 0.25) - a handler is triggered

□ 0.25) - this handler provides a response page with all events

□ 0.5) - the response page displays in tabular format



Task - 3: List sold out events (2m) *

○ 0.0) - insufficient evidence

○ 0.5) - a handler is triggered

○ 1.0) - this handler provides a response page with all categories

○ 2.0) - the data is listed in a tabular format

Task - 4: Category Detail Page (3m) *

□ 0.0) - insufficient evidence

□ 0.25) - correct method is declared

□ 0.25) - the pathname is triggered

□ 0.5) - category ID is retrieved correctly

□ 1.0) - the page show the category details and its events

□ 1.0) - the response in a tabular format

Task - 5: Delete Event By ID (2m) *

□ 0.0) - insufficient evidence

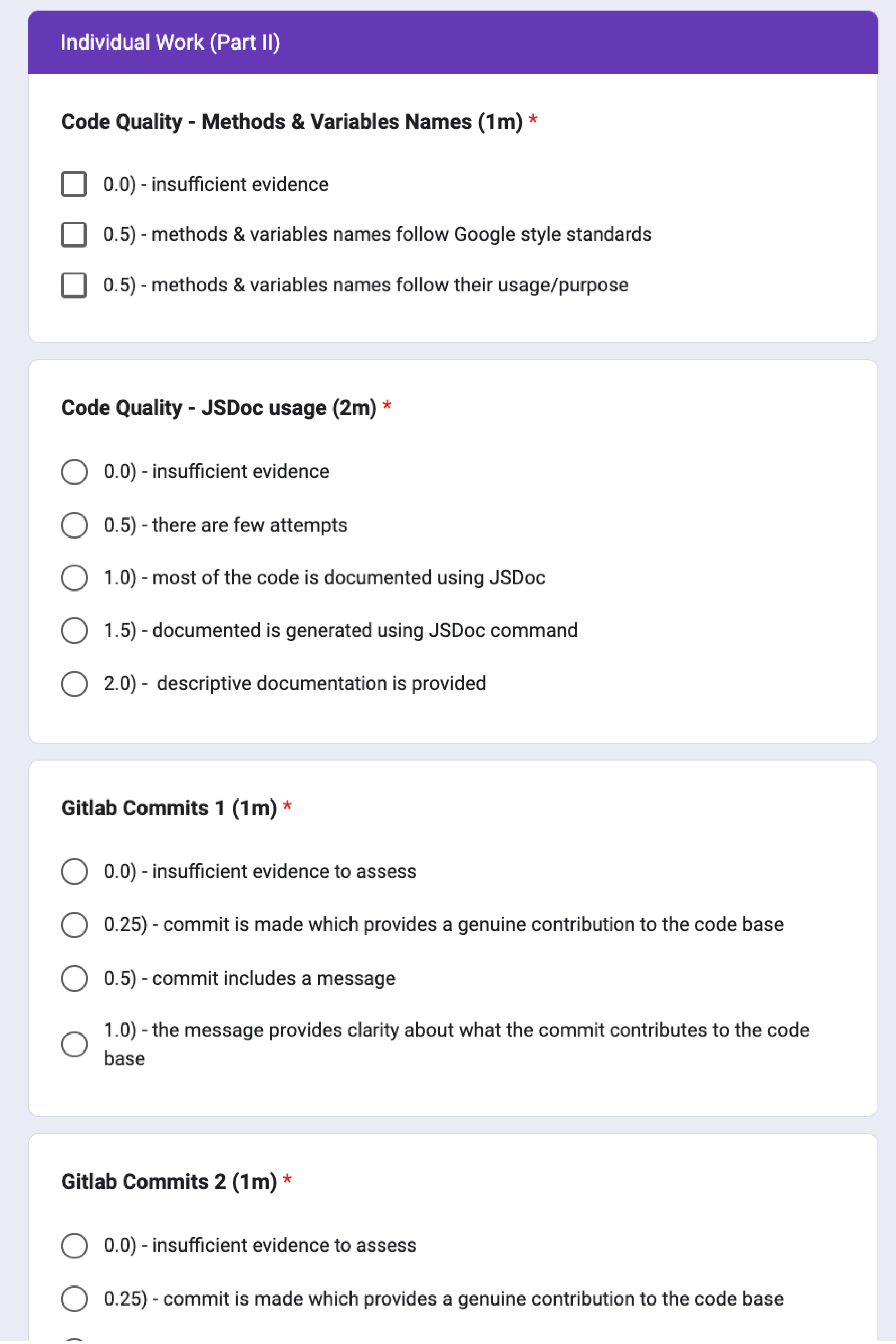
□ 0.5) - correct HTTP method is used

□ 0.5) - the ID is retrieved correctly

□ 0.5) - the event is deleted correctly

□ 0.5) - the endpoint responds back to the client

Individual Assessments (All students)



Individual Work (Part II)

Code Quality - Methods & Variables Names (1m) *

□ 0.0) - insufficient evidence

□ 0.5) - methods & variables names follow Google style standards

□ 0.5) - methods & variables names follow their usage/purpose

Code Quality - JSDoc usage (2m) *

○ 0.0) - insufficient evidence

○ 0.5) - there are few attempts

○ 1.0) - most of the code is documented using JSDoc

○ 1.5) - documented is generated using JSDoc command

○ 2.0) - descriptive documentation is provided

Gitlab Commits 1 (1m) *

○ 0.0) - insufficient evidence to assess

○ 0.25) - commit is made which provides a genuine contribution to the code base

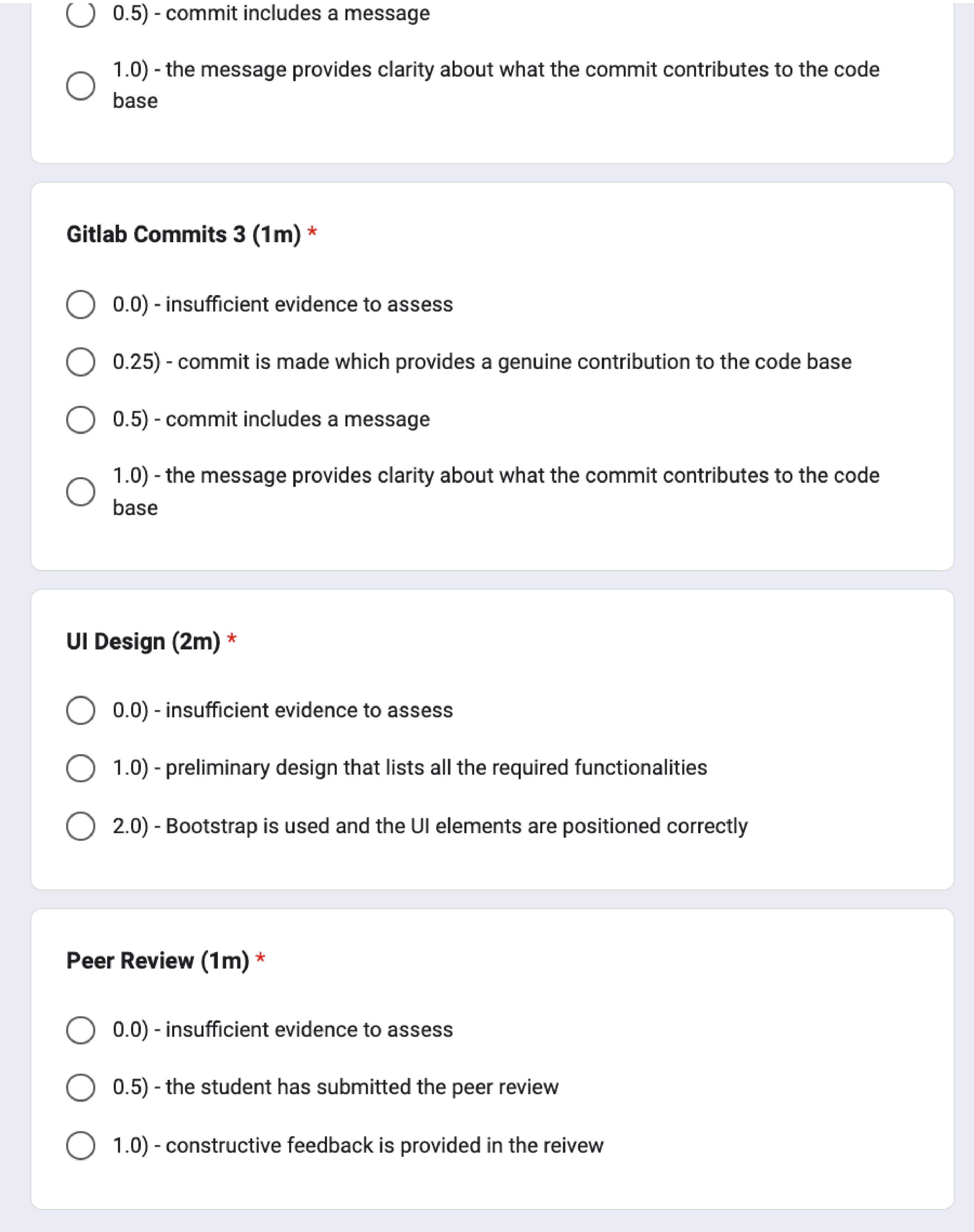
○ 0.5) - commit includes a message

○ 1.0) - the message provides clarity about what the commit contributes to the code base

Gitlab Commits 2 (1m) *

○ 0.0) - insufficient evidence to assess

○ 0.25) - commit is made which provides a genuine contribution to the code base



○ 0.5) - commit includes a message

○ 1.0) - the message provides clarity about what the commit contributes to the code base

Gitlab Commits 3 (1m) *

○ 0.0) - insufficient evidence to assess

○ 0.25) - commit is made which provides a genuine contribution to the code base

○ 0.5) - commit includes a message

○ 1.0) - the message provides clarity about what the commit contributes to the code base

UI Design (2m) *

○ 0.0) - insufficient evidence to assess

○ 1.0) - preliminary design that lists all the required functionalities

○ 2.0) - Bootstrap is used and the UI elements are positioned correctly

Peer Review (1m) *

○ 0.0) - insufficient evidence to assess

○ 0.5) - the student has submitted the peer review

○ 1.0) - constructive feedback is provided in the reivew

Where & What to submit? 📤

Where?

Every student must submit assignment-1 to two places: **Moodle** and **Gitlab**.

For Moodle, you must ZIP your files into a single file and upload it to the assignment-1 submission link that can be found on Moodle-->Assessment. There is a quick survey you must do to enable the submission link.

For GitLab, you will be provided with a group repository, and you should commit/push your project to it. We are expecting at least three non-trivial commits to your group repo. See the marking rubric for more details.

> ⚠ We will mark your assignment based on the latest commit in the `main` branch in your GitLab repository by the due date, **not** in the `master` branch.

What??

1. A document in PDF, DOCX, or MD format that contains:
    1. The routing table that is requested in the Assignment-1 specification
    2. Steps to run your applications
    3. [Optional] unsolved bugs and issues
    4. this document must be placed in the root folder of your project
2. The source code of the project

Late Submission

References and Support 🆘

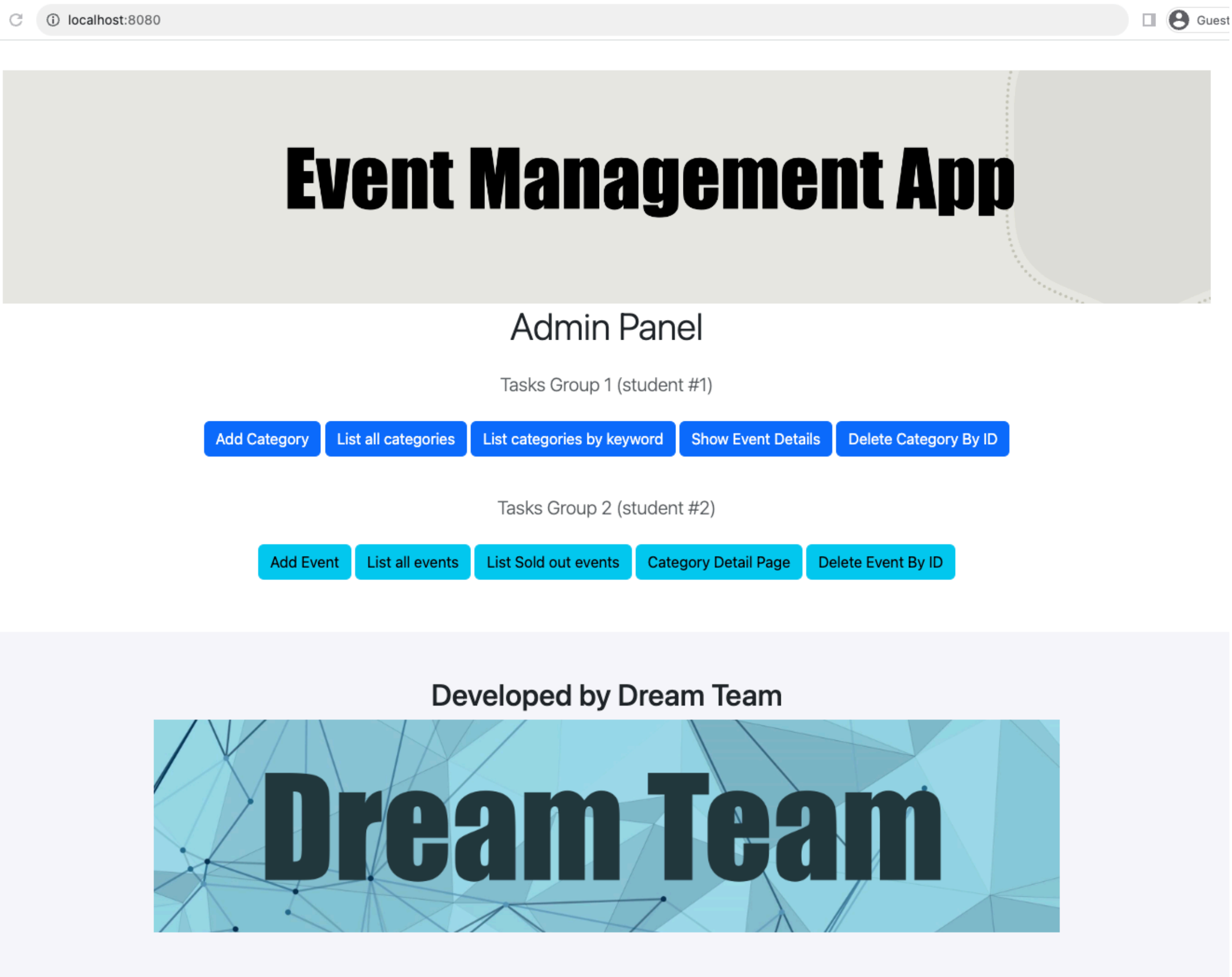
- Generate Random numbers in JS and Node.js
- Arrays in JavaScript/Node.js
- Classes in JavaScript/Node.js
- Slice an Array in JS/Node.js

Sample Screenshots

⚠ **For reference only**

Please use these screenshots for reference only, your expected output should align as per assignment specifications. Feel free to be creative as EMA is your product.

Home Page



localhost:8080

Event Management App

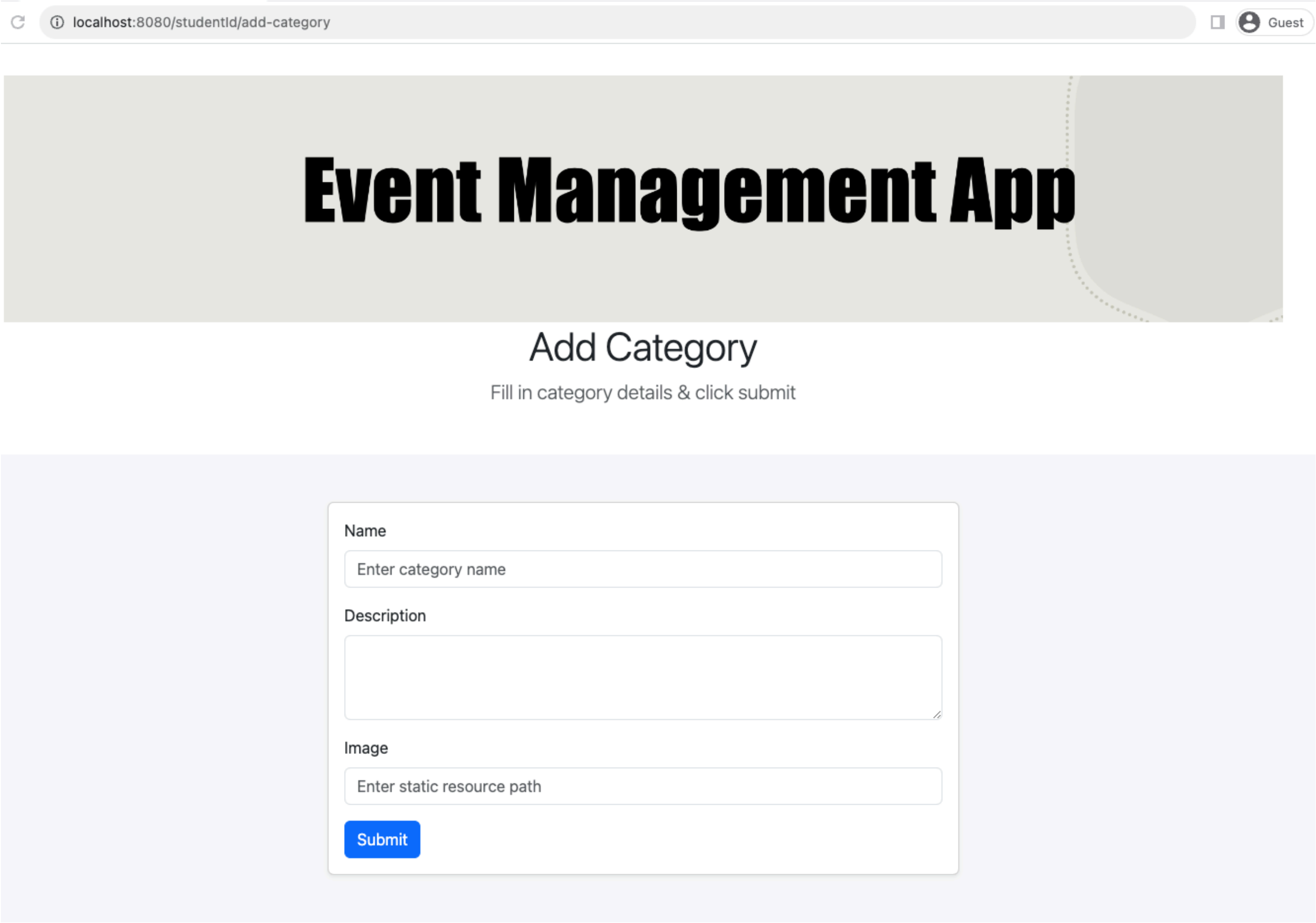
Admin Panel

Tasks Group 1 (student #1)

Tasks Group 2 (student #2)

Developed by Dream Team

Add Category



localhost:8080/studentId/add-category

Event Management App

Add Category

Fill in category details & click submit

Name

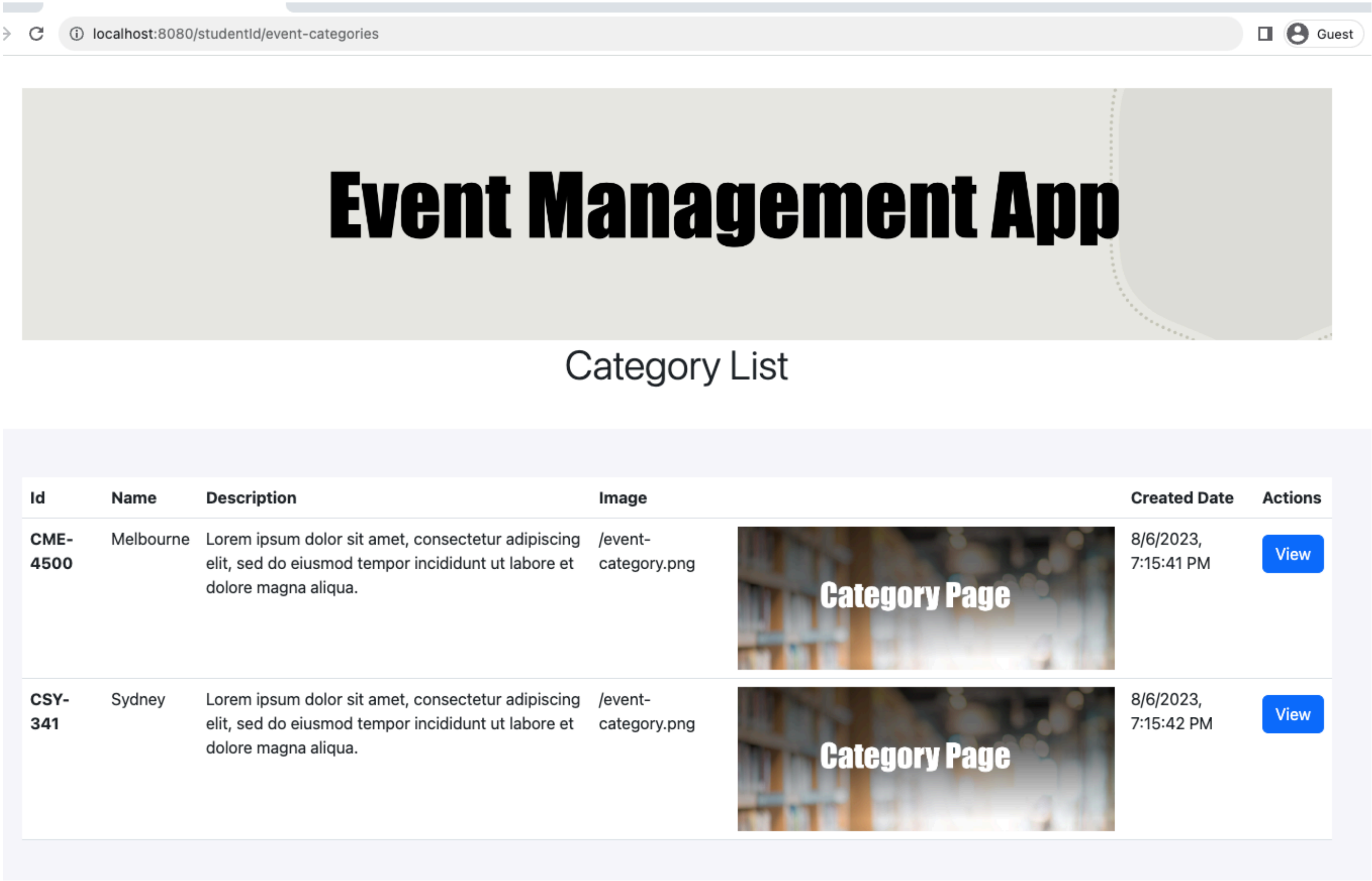
Enter category name

Description

Image

Enter static resource path

List Categories

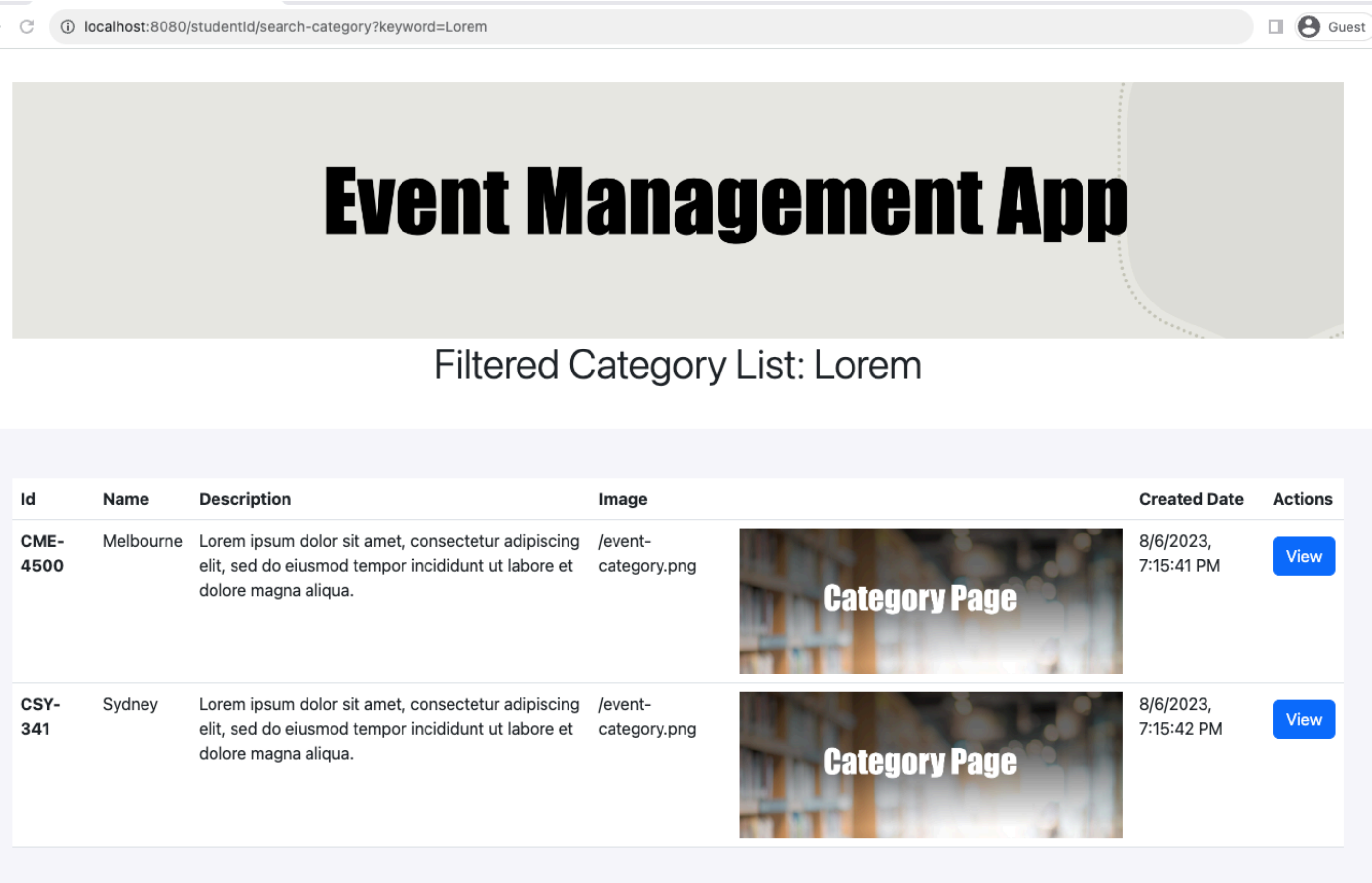


Event Management App

Category List

Id	Name	Description	Image		Created Date	Actions
CME-4500	Melbourne	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	/event-category.png	Category Page	8/6/2023, 7:15:41 PM	View
CSY-341	Sydney	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	/event-category.png	Category Page	8/6/2023, 7:15:42 PM	View

List categories by keyword

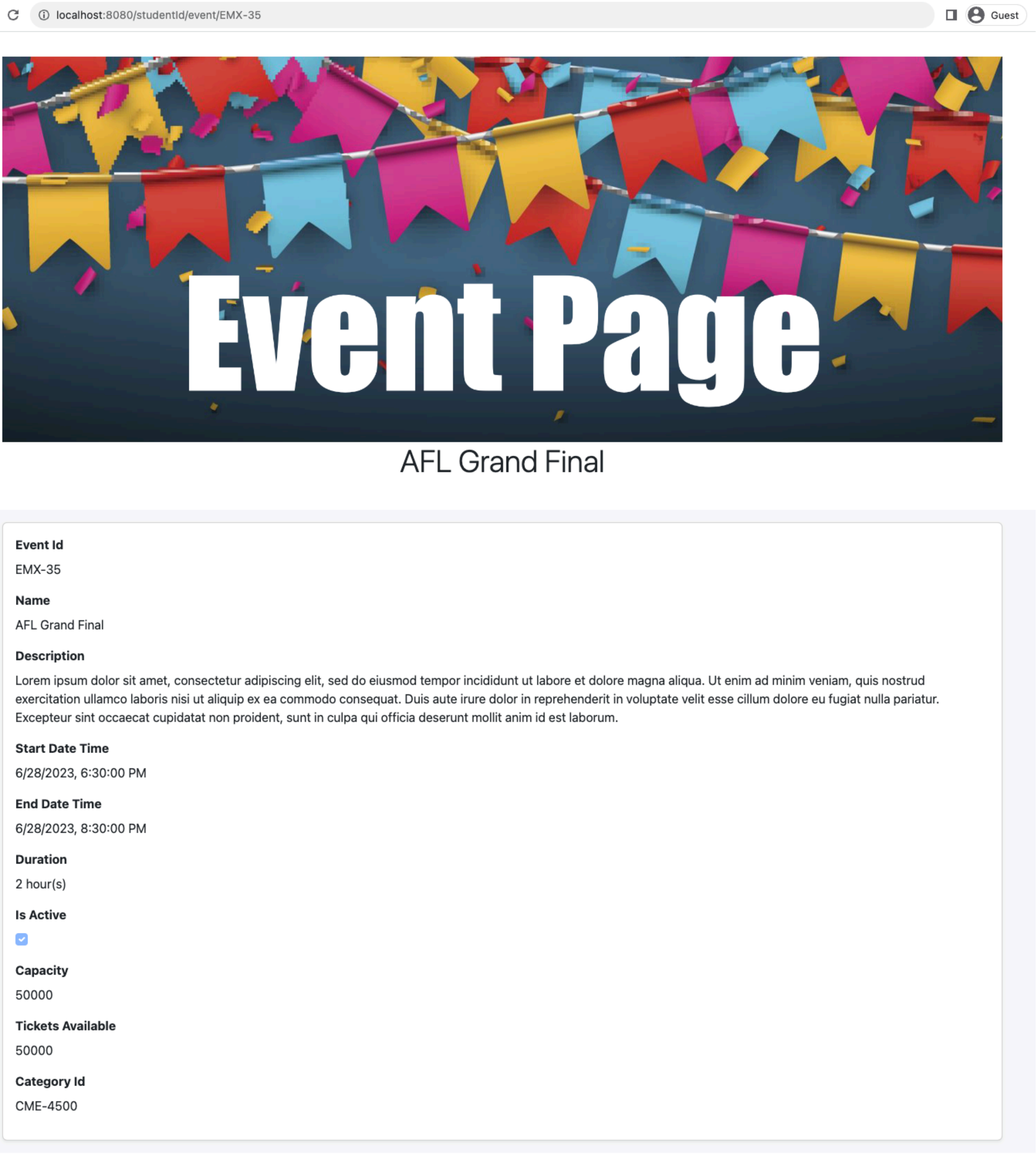


Event Management App

Filtered Category List: Lorem

Id	Name	Description	Image		Created Date	Actions
CME-4500	Melbourne	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	/event-category.png	Category Page	8/6/2023, 7:15:41 PM	View
CSY-341	Sydney	Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.	/event-category.png	Category Page	8/6/2023, 7:15:42 PM	View

Show Event Details



AFL Grand Final

Event Id

EMX-35

Name

AFL Grand Final

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Start Date Time

6/28/2023, 6:30:00 PM

End Date Time

6/28/2023, 8:30:00 PM

Duration

2 hour(s)

Is Active

☑

Capacity

50000

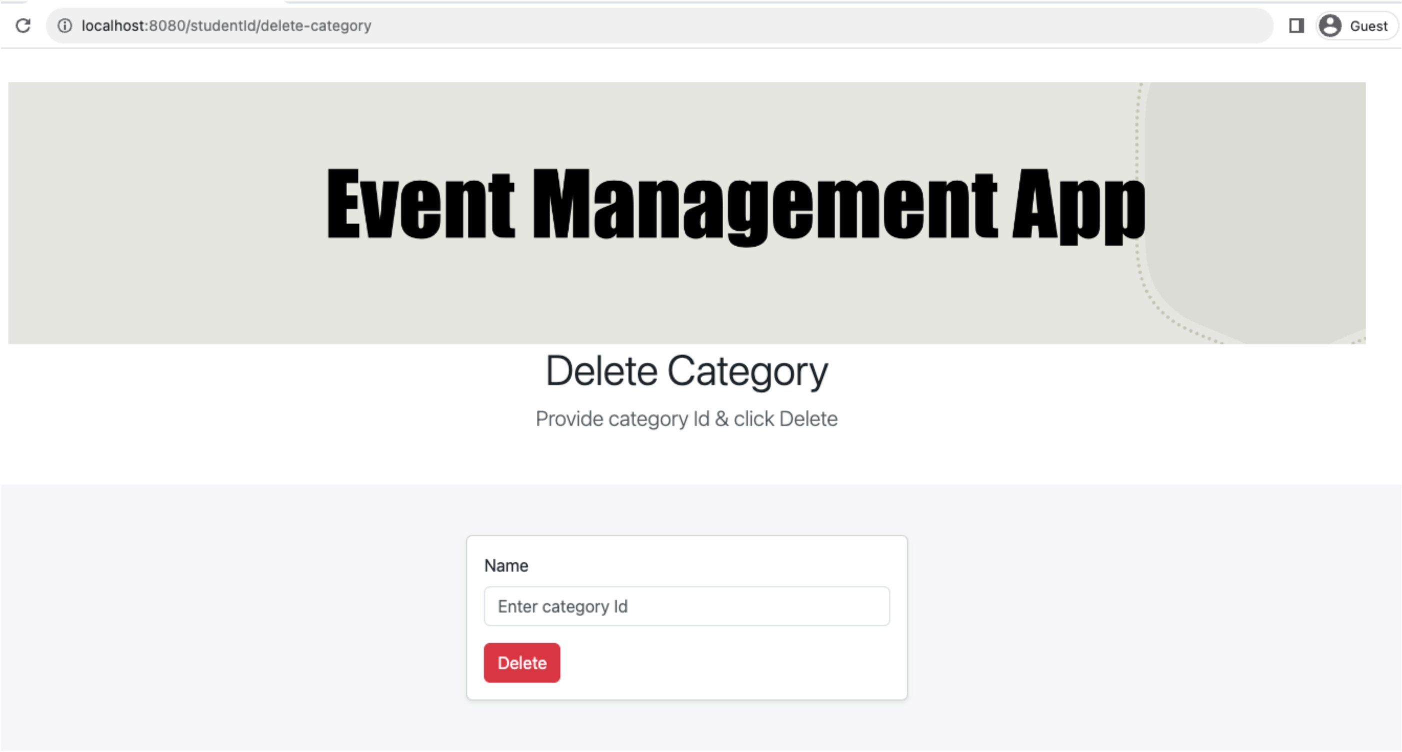
Tickets Available

50000

Category Id

CME-4500

Delete Category



localhost:8080/studentId/delete-category

Event Management App

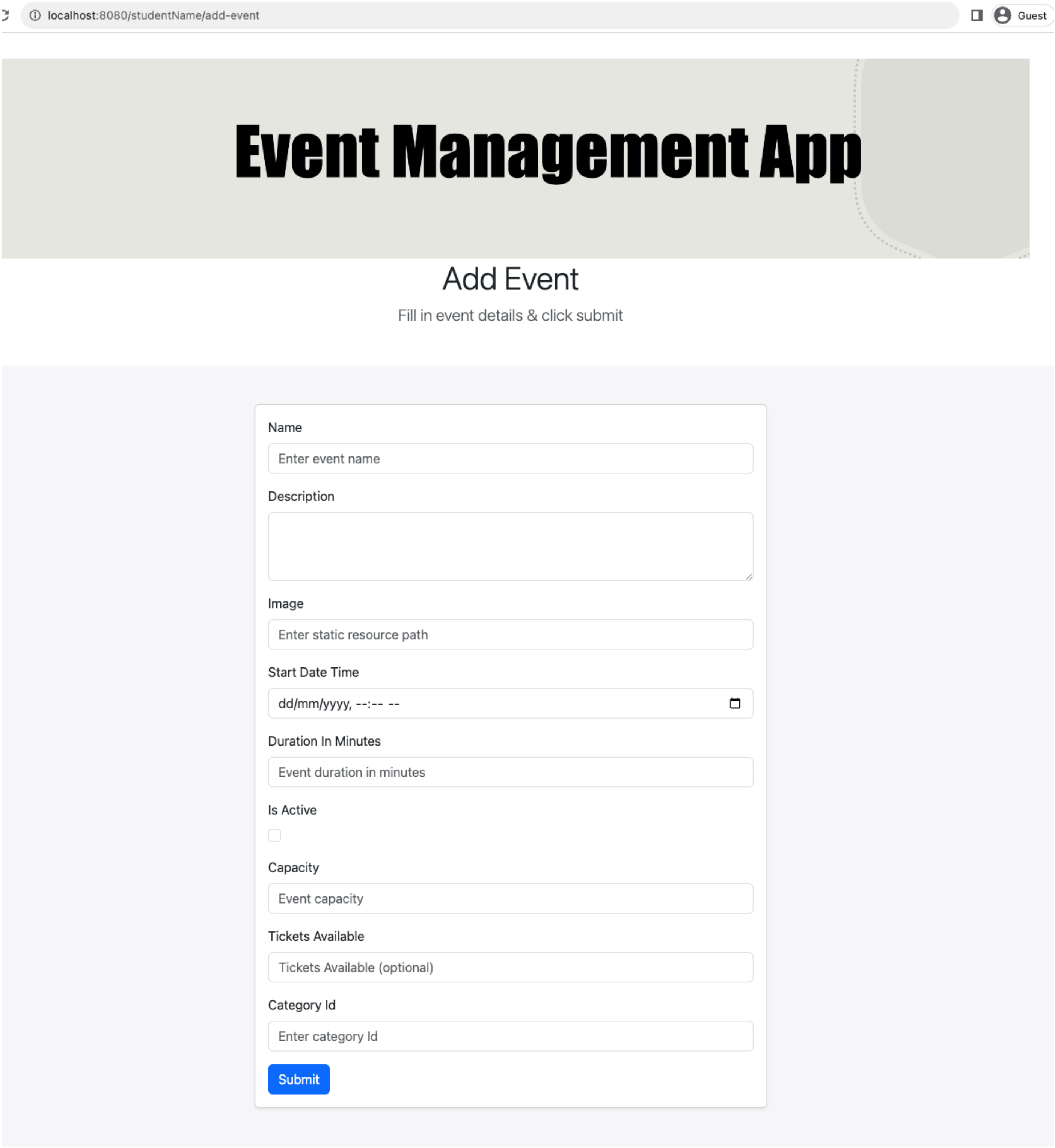
Delete Category

Provide category Id & click Delete

Name

Enter category Id

Add Event Form



localhost:8080/studentName/add-event

Event Management App

Add Event

Fill in event details & click submit

Name

Enter event name

Description

Image

Enter static resource path

Start Date Time

dd/mm/yyyy, --:-- --

Duration In Minutes

Event duration in minutes

Is Active

□

Capacity

Event capacity

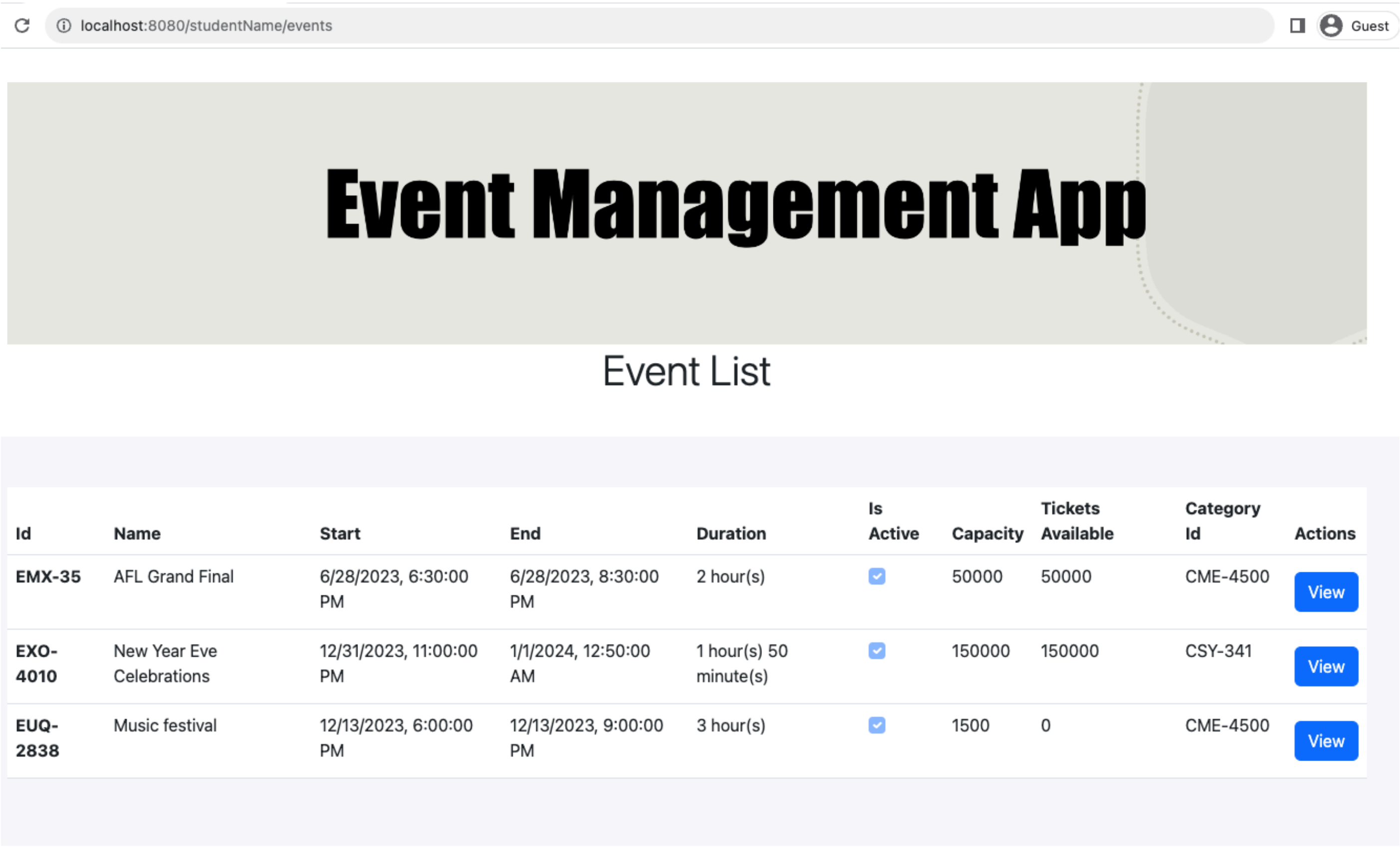
Tickets Available

Tickets Available (optional)

Category Id

Enter category Id

List all events

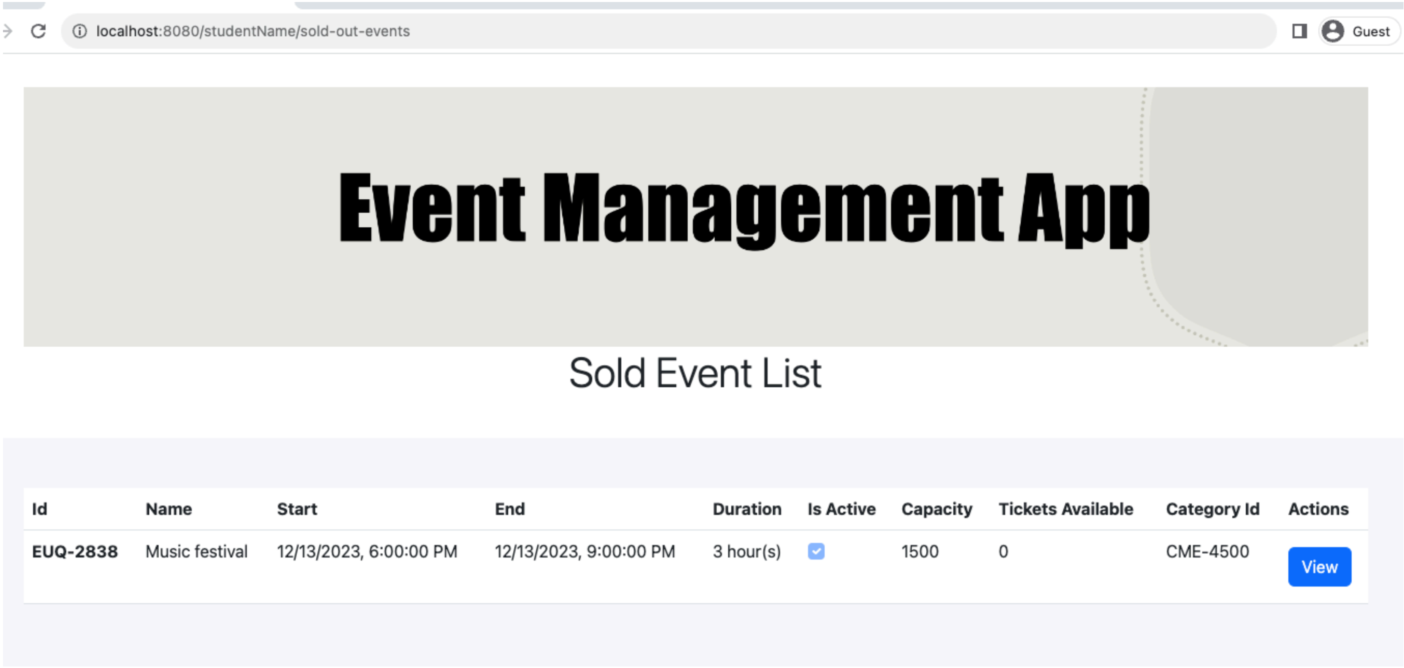


Event Management App

Event List

Id	Name	Start	End	Duration	Is Active	Capacity	Tickets Available	Category Id	Actions
EMX-35	AFL Grand Final	6/28/2023, 6:30:00 PM	6/28/2023, 8:30:00 PM	2 hour(s)	☑	50000	50000	CME-4500	View
EXO-4010	New Year Eve Celebrations	12/31/2023, 11:00:00 PM	1/1/2024, 12:50:00 AM	1 hour(s) 50 minute(s)	☑	150000	150000	CSY-341	View
EUQ-2838	Music festival	12/13/2023, 6:00:00 PM	12/13/2023, 9:00:00 PM	3 hour(s)	☑	1500	0	CME-4500	View

List Sold out events

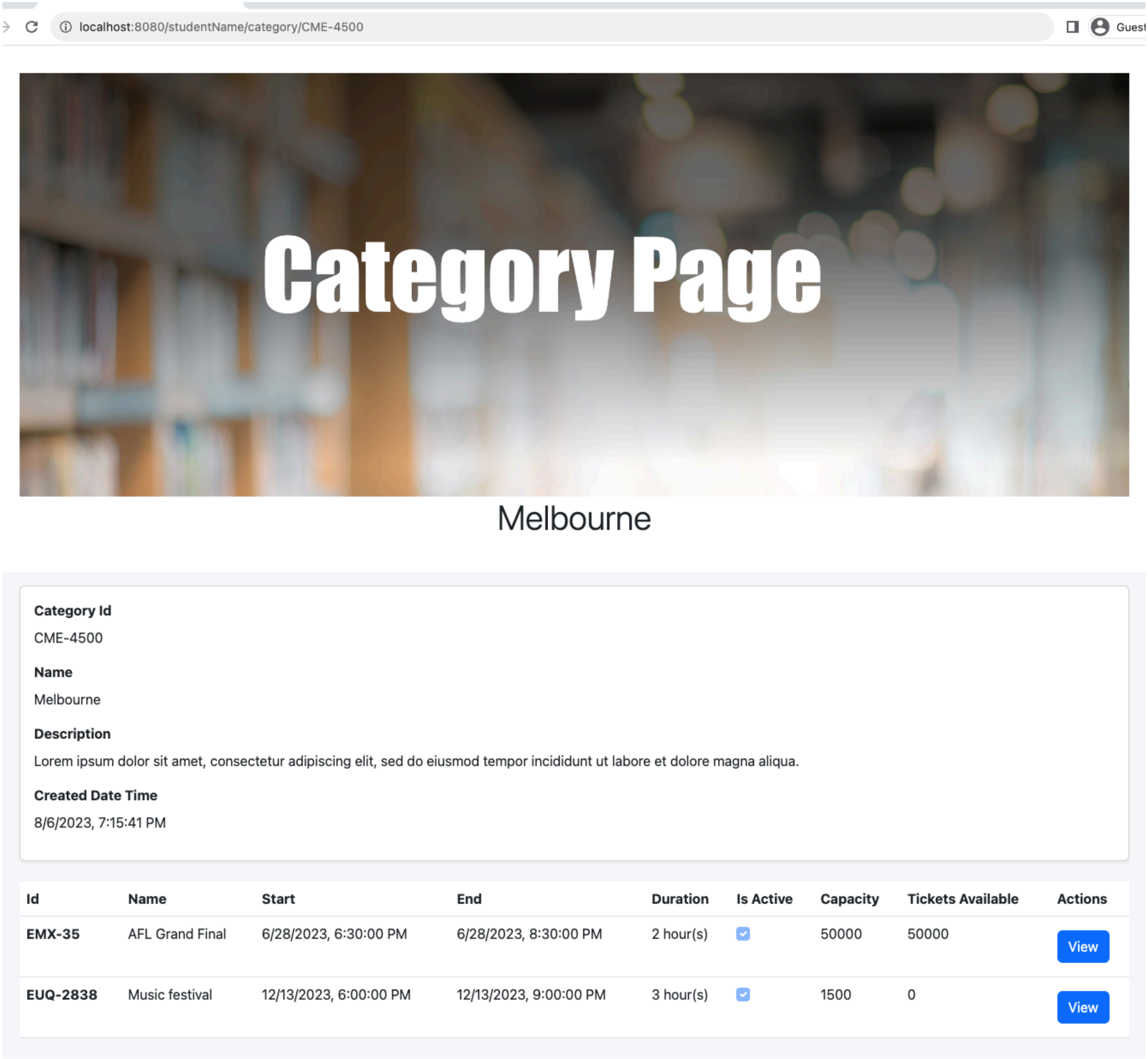


Event Management App

Sold Event List

Id	Name	Start	End	Duration	Is Active	Capacity	Tickets Available	Category Id	Actions
EUQ-2838	Music festival	12/13/2023, 6:00:00 PM	12/13/2023, 9:00:00 PM	3 hour(s)	☑	1500	0	CME-4500	View

Category Detail Page



localhost:8080/studentName/category/CME-4500

Category Page

Melbourne

Category Id

CME-4500

Name

Melbourne

Description

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Created Date Time

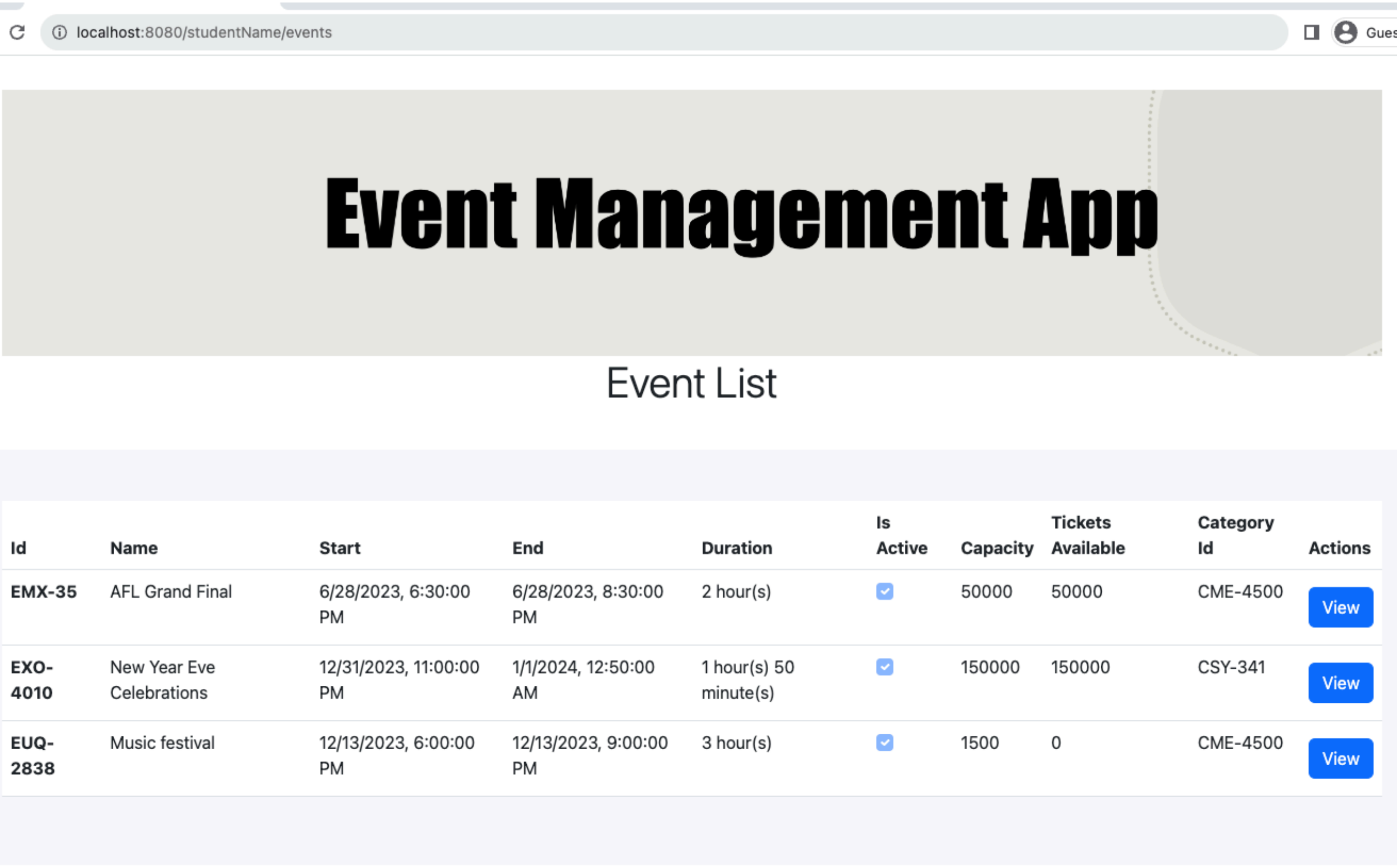
8/6/2023, 7:15:41 PM

Id	Name	Start	End	Duration	Is Active	Capacity	Tickets Available	Actions
EMX-35	AFL Grand Final	6/28/2023, 6:30:00 PM	6/28/2023, 8:30:00 PM	2 hour(s)	☑	50000	50000	View
EUQ-2838	Music festival	12/13/2023, 6:00:00 PM	12/13/2023, 9:00:00 PM	3 hour(s)	☑	1500	0	View

Delete Event by Id

Example: http://localhost:8080/studentName/delete-event?eventId=EMH-2107

and the event will be deleted from the DB, and the user will be redirected to all events pages, as shown below.



Event Management App

Event List

Id	Name	Start	End	Duration	Is Active	Capacity	Tickets Available	Category Id	Actions
EMX-35	AFL Grand Final	6/28/2023, 6:30:00 PM	6/28/2023, 8:30:00 PM	2 hour(s)	☑	50000	50000	CME-4500	View
EXO-4010	New Year Eve Celebrations	12/31/2023, 11:00:00 PM	1/1/2024, 12:50:00 AM	1 hour(s) 50 minute(s)	☑	150000	150000	CSY-341	View
EUQ-2838	Music festival	12/13/2023, 6:00:00 PM	12/13/2023, 9:00:00 PM	3 hour(s)	☑	1500	0	CME-4500	View

How to Clone your Repo?

In this recording, I demonstrate how to clone your assignment 1 repository to your machine and then commit and push changes back to the remote repo.