

Test Project

Module C (Back-end & Admin)
Web Technologies

Submitted by:

Oscar Fernando Aristizábal Cardona
National Expert Colombia

Table of Contents

Table of Contents	3
Introduction	4
Project Description and Tasks to be Performed.....	4
Parte 1: API REST (Backend).....	4
Endpoints of the API and Possible Responses:	4
DataBase:.....	5
Login	6
Part 2: Administration Panel (Frontend)	6
1. Login:	6
2. Events Management:	7
3. Create Event:	7
4. Create Participant:	8
5. Edit Event:	8
6. List of Participants by Event:	9
Instructions for the Competitor:	10
Marking Scheme	10

Introduction

The French Olympic Committee is looking to develop a comprehensive system to manage and coordinate events, participants, and venues for the upcoming Olympics in France.

This project involves creating a high-performance **REST API (Backend)** and developing a sophisticated administration panel (**Frontend**). This full-stack system will handle event management, participant coordination, venue management, and other essential aspects of Olympic operations, enabling efficient and effective administration of all related activities.

Project Description and Tasks to be Performed

The time allocated for this test is 3 hours, and the database structure will be provided in this document. You will need to design the necessary tables and relationships to store, query, edit, and delete information as required. The project must meet security and data protection standards, as it deals with sensitive information related to events and participants.

Parte 1: API REST (Backend)

For constructing the REST API, you can use two popular technologies:

- **PHP/Laravel**
- **Node.js/Express**

Endpoints of the API and Possible Responses:

The configuration or definition of the endpoints must have the prefix 'olympics' and respect the possible responses designated below for each functionality:

1. Administrator Authentication:

- **Endpoint:** /olympics/login ¹
 - 200 OK: Successful authentication.
 - 401 Unauthorized: Authentication failed.
- **Endpoint:** /olympics/logout ²
 - 200 OK: Successful logout.


2. Events Registration:

- **Endpoint:** /olympics/events/create ³
 - 201 Created: Successful event registration.
 - 400 Bad Request: Request error.


3. Show Registered Events:

- **Endpoint:** /olympics/events/list ⁴
 - 200 OK: Successful query.
 - 404 Not Found: Resource not found


4. Edit Registered Events:

- **Endpoint:** /olympics/events/edit/{idevent} 
 - 200 OK: Successful edit.
 - 404 Not Found: Event not found.
 - 400 Bad Request: Request error.


5. Participant Registration:

- **Endpoint:** /olympics/participants/create 
 - 201 Created: Successful participant registration.
 - 400 Bad Request: Request error.

6. Show Participants by Event:

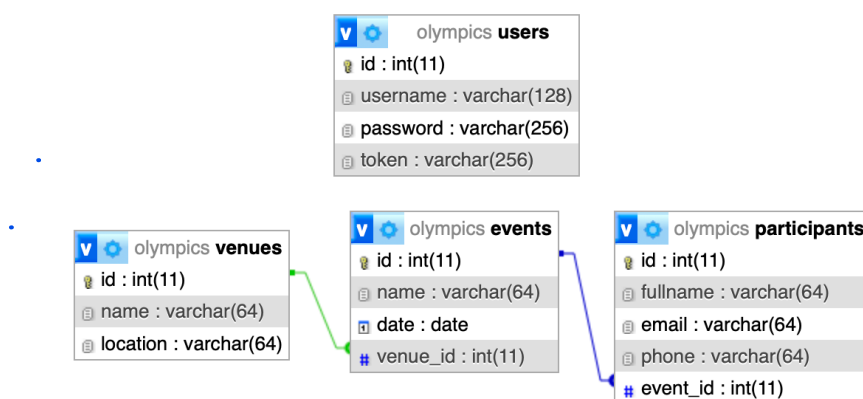
- **Endpoint:** /olympics/participants/list/{idevent} 
 - 200 OK: Successful query.
 - 404 Not Found: Resource not found.

7. Delete Participant Data:

- **Endpoint:** /olympics/participants/delete/{id} 
 - 204 No Content: Successful deletion.
 - 404 Not Found: Participant not found.
 - 400 Bad Request: Request error.

DataBase:

The provided database structure must be adhered to:



Since the data in the **'venues'** table is fixed, these do not need to be entered from the administration panel. Please register them in the database.

Data (Venues):

ID	NAME	LOCATION
1	Stade de France	Saint-Denis
2	Arena Bercy	París
3	Roland Garros	París

4	Vélodrome National	Saint-Quentin
5	Grand Palais	Paris

Login

It is crucial to implement an authentication system with username and password in the administration panel. The **'/login'** endpoint must generate a random security token. All API calls from the administration panel must include this token in the authorization header.

This process ensures that only authorized users access and operate the system, protecting the data and functions of the API and the administration interface.

When creating users, they must be stored in the **'users'** table, and one user must be registered with the following information:

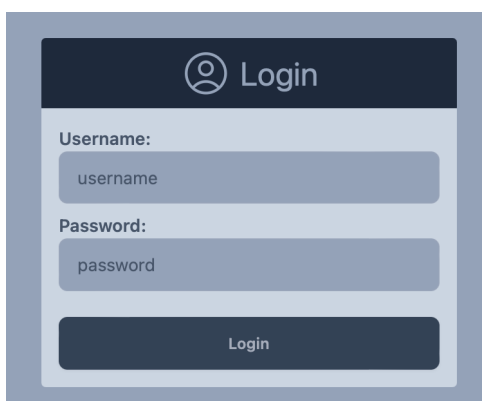
- **username:** admin
- **password:** admin

Part 2: Administration Panel (Frontend)

To build the administration interfaces, you have various JavaScript framework options like React or Vue.js. These frameworks provide an organized structure and rapid user interface development.

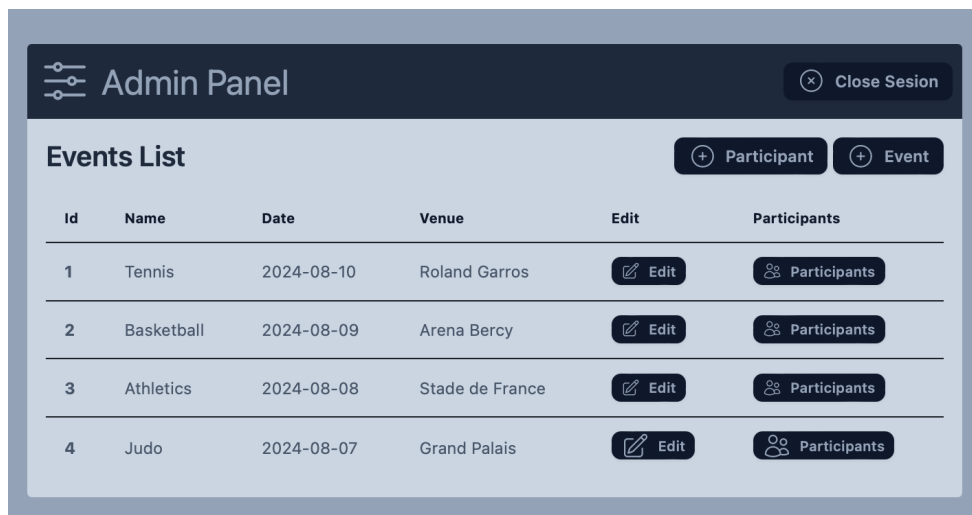
The administration panel must include the following essential functionalities, ensuring the navigation maintains the flow described below:

1. Login:


 A mockup of a login form. It has a dark header bar with a user icon and the word "Login". Below the header, there are two input fields: "Username:" with a placeholder "username" and "Password:" with a placeholder "password". At the bottom of the form is a dark button labeled "Login".

- Administrators must authenticate using a username and password.
- If the entered information does not match the database, access should not be allowed, and an appropriate message should be displayed.
- If the information matches, the user should be redirected to the **"Event Management"** page.

2. Events Management:

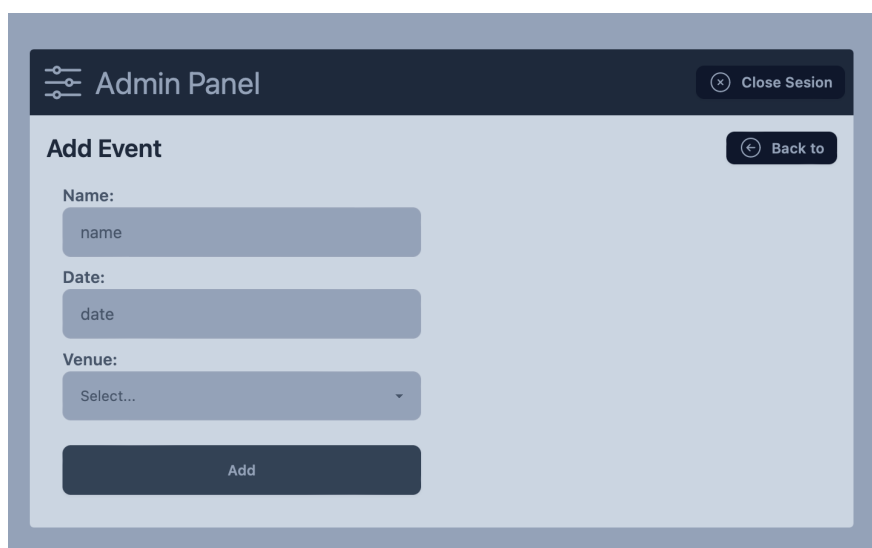


The screenshot shows the 'Admin Panel' with a 'Close Sesion' button in the top right. Below the header is the 'Events List' section, which includes two buttons: '+ Participant' and '+ Event'. The main content is a table with the following data:

Id	Name	Date	Venue	Edit	Participants
1	Tennis	2024-08-10	Roland Garros	Edit	Participants
2	Basketball	2024-08-09	Arena Bercy	Edit	Participants
3	Athletics	2024-08-08	Stade de France	Edit	Participants
4	Judo	2024-08-07	Grand Palais	Edit	Participants

- A **"Close Sesion"** option must appear at the top right of all interfaces, redirecting to the login page and destroying the session or token. ✓
- The table should display registered event information in chronological order. ✓
- Clicking the **" + Event "** button should display a form to register event information. ✓
- Clicking the **" + Participant "** button should display a form to register participant information. ✓
- The **"Edit"** button should load the event information to modify its date and venue. ✓
- The **"Participants"** button should load and display the records of participants belonging to that event. ✓

3. Create Event:



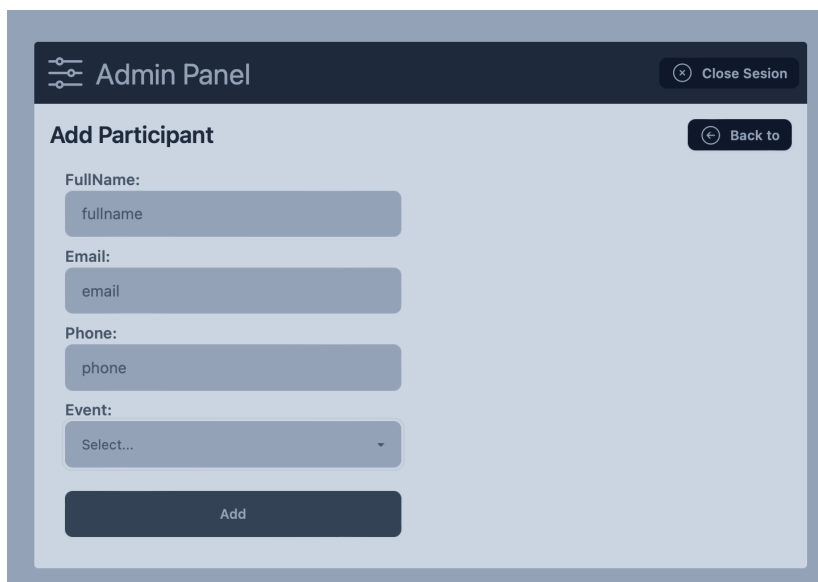
The screenshot shows the 'Admin Panel' with a 'Close Sesion' button in the top right. Below the header is the 'Add Event' section, which includes a 'Back to' button. The form contains the following fields:

- Name:
- Date:
- Venue:

At the bottom of the form is an 'Add' button.

- In the event registration form, event data must be entered and displayed on the Event Management page.
- All information is mandatory, so registration should not be allowed if any data is missing.

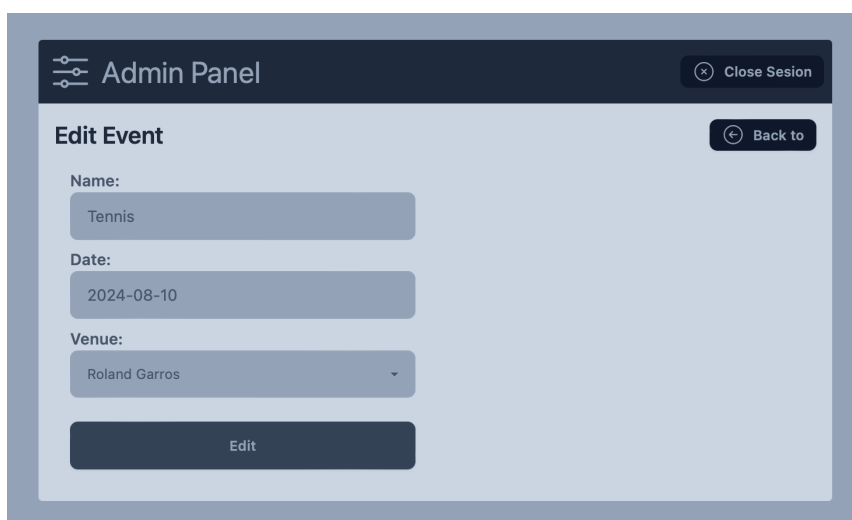
4. Create Participant:



The screenshot shows the 'Add Participant' form within the 'Admin Panel'. The form has a dark header with a menu icon and the text 'Admin Panel', and a 'Close Sesion' button. The main content area is titled 'Add Participant' and includes a 'Back to' button. The form fields are: 'FullName' (text input), 'Email' (text input), 'Phone' (text input), and 'Event' (dropdown menu with 'Select...' as the placeholder). An 'Add' button is located at the bottom of the form.

- In the participant registration form, personal participant information must be registered, and the event information should be shown in the "**Event**" selection field.
- All information is mandatory.

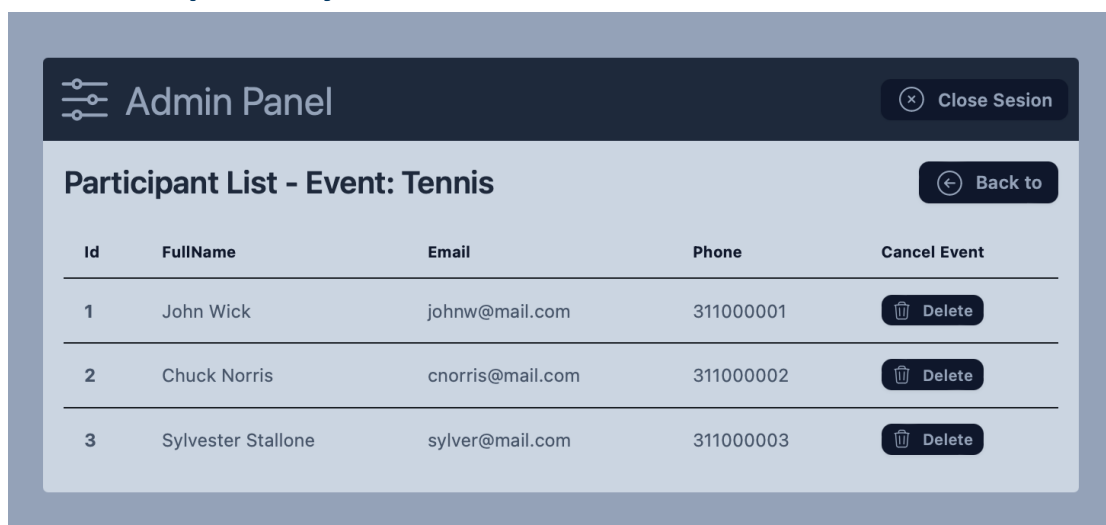
5. Edit Event:



The screenshot shows the 'Edit Event' form within the 'Admin Panel'. The form has a dark header with a menu icon and the text 'Admin Panel', and a 'Close Sesion' button. The main content area is titled 'Edit Event' and includes a 'Back to' button. The form fields are: 'Name' (text input with 'Tennis' as the value), 'Date' (text input with '2024-08-10' as the value), and 'Venue' (dropdown menu with 'Roland Garros' as the selected value). An 'Edit' button is located at the bottom of the form.

- The only information that can be modified and updated for an event once registered is its date; all other data should be in read-only mode.

6. List of Participants by Event:



- The list of participants by event should display all participant information records in alphabetical order by their name.
- Clicking the "Delete" button should remove the participant's information from the database, and the interface should redirect to itself.

Notes:

- When registering, updating, and/or deleting data, the interface should redirect to the current screen.
- Try to navigate between the interfaces, considering the logic of the different administration panel interfaces, or create a readme that indicates the routes to access the administration panel interfaces.
- The generated interfaces do not need to resemble the provided interface designs in terms of design; it is only necessary to respect the order of the fields in forms and tables.

Instructions for the Competitor:

Follow these instructions to submit your work:

1. Save and work on your API in a directory in your '**htdocs**' called '**moduleC-XYZ-API**', where '**XYZ**' should be replaced with the name of your region.
2. Save and work on your administration panel in a directory in your '**htdocs**' called '**moduleC-XYZ-admin**', where '**XYZ**' should be replaced with the name of your region.

Marking Scheme

SECTION	CRITERION	JUDGEMENT	MEASUREMENT	TOTAL
C1	Login (API)	0	2	2
C2	Evento (API)	0	6	6
C3	Participants (API)	0	5	5
C4	General Aspects (API)	1	2	3
C5	Login (Admin)	0	3.5	3.5
C6	Events Management (Admin)	0	2.5	2.5
C7	Participants Management	0.5	2.5	3
		1.5	23.5	25

16
9