

Test Project

Web Technologies

PHP and JavaScript Module

Submitted by:

Sebastian Häni – CH (Independent Test Project Designer)

Cyril Wanner – CH (Independent Test Project Designer)

Patrick Zahnd – CH (Independent Test Project Designer)

Contents

Contents	2
Introduction	3
Description of project and tasks	3
Phase one – PHP	4
Database design.....	4
Organizer interface	4
<i>Organizer login and logout</i>	<i>5</i>
<i>Event management</i>	<i>6</i>
<i>Ticket management.....</i>	<i>9</i>
<i>Session management.....</i>	<i>10</i>
<i>Channel management</i>	<i>12</i>
<i>Room management.....</i>	<i>13</i>
<i>Room capacity.....</i>	<i>14</i>
Attendee API.....	15
<i>Events overview</i>	<i>15</i>
<i>Event detail.....</i>	<i>16</i>
<i>Attendee login and logout.....</i>	<i>17</i>
<i>Event registration</i>	<i>18</i>
Phase two – JavaScript	19
API Setup	19
Introduction	20
Events list.....	21
Event agenda.....	22
Session detail	23
Event registration	24
Attendee login	26
Notes.....	27
Submission Instructions to the Competitor	28
Equipment, machinery, installations, and materials required.....	29
Marking Scheme	29

Introduction

The agency responsible for organizing the WorldSkills Conferences needs a new web application helping them to manage the events and allowing attendees to sign up and see all information they need. It should be dynamically built so that it can later be shared with other member organizations which means that multitenancy is an important requirement.

Because this application will grow over time, important requirements are covered by automated tests.

Description of project and tasks

The project is split into two phases:

- phase one (morning, 3 hrs) for building the backend parts (admin interface and REST-API) using PHP (framework) and database
- phase two (afternoon, 3 hrs) for building the frontend parts (attendee interface) using HTML/CSS/JS (framework) and the REST-API provided for phase two.

Phase one – PHP

Database design

- Use your database (dbYY_competitor_day2) and import the schema/dump provided
- Extend your database to be able to store (do not modify/extend any existing table except if it is specifically noted as they will be resetted before every test run)
 - Event ratings
 - Attendees should be able to comment and rate events they have attended
 - Rating should be a full number between 1 and 5, where 1 is the lowest and 5 the highest rating
 - The date and time of the rating should be registered
 - Session ratings
 - Additional to the event rating, attendees should be able to rate sessions they have attended
 - Rating should be a full number between 1 and 5, where 1 is the lowest and 5 the highest rating
 - The date and time of the rating should be registered
- Your additional tables need to follow the third normal form.

Organizer interface

Create an organizer interface (Server Side Rendered using a PHP Framework) using the templates provided to

- manage events, sessions, channels, rooms and tickets (create, read, update)
- create a bar chart to see the occupancy rates of all rooms
- create dynamic ticket prices
- support multiple tenants

Because different organizers will use this backend, keep security in mind. It should not be possible to access events of other organizers or update events of other organizers.

Organizer login and logout

Because registering as a new organizer is not within the scope of this project, please update the first two organizers in the provided dump so that it is possible to log in with the following credentials:

- Email: demo1@worldskills.org Password: demopass1
- Email: demo2@worldskills.org Password: demopass2

WorldSkills Event Platform

Please sign in

Email
Password

Sign in

Specification:

Feature: A1 – Log in and log out as an organizer: As an organizer, I would like to log in and log out to the event platform so that I can access the functionality of the platform.

Scenario: Ala – Organizer login ok

Given I am on the "login" page

When I enter the specified email and password for an organizer

And press the "Sign in" button

Then I am forwarded to the organizer area

And organizer name and "Sign out" button become visible in the header

And page "Manage events" becomes visible

Scenario: Alb – Organizer login fail

Given I am on the "login" page

When I enter a wrong email and password combination for an organizer
or use login information of an attendee instead of an organizer

And press the "Sign in" button

Then I stay on the "login" page

And I see an error message "Email or password not correct"

Scenario: Alc – Organizer logout

Given I am logged in

When I press the "Sign out" button

Then I am forwarded to the "login" page

And organizer name and logout button and admin functions disappear

And I cannot reach the previous page by going back

Event management

Organizers should be able to save all information about an event and its sessions in the database. Every organizer account can only see and edit its own events. The organizer may have zero or many events.

Specification:

Feature: A2 - Manage events: As an organizer I want to manage events so attendees can find events

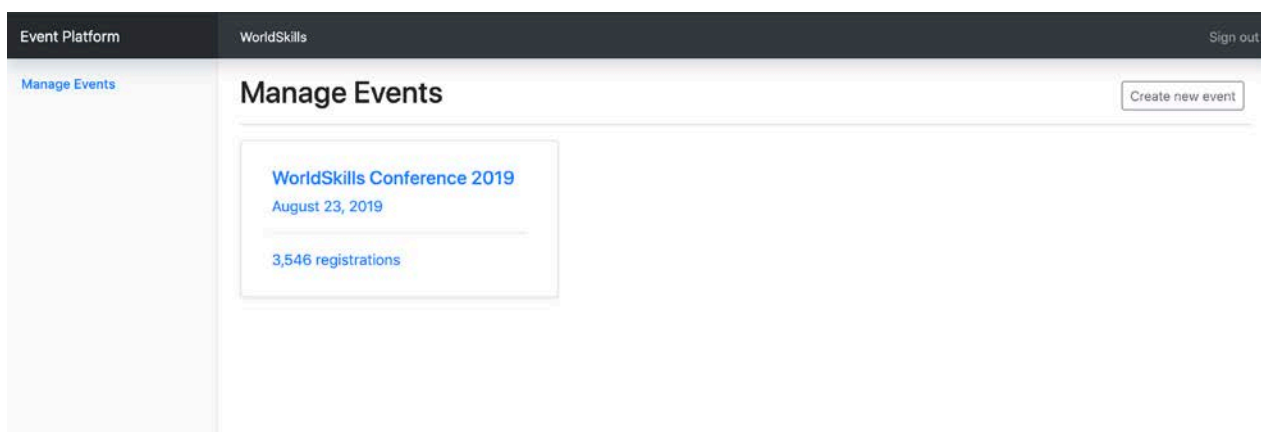
Scenario: A2a - List all events

Given I am logged in

And I am on the "Manage events" page

Then I see a list of all events created by the current organizer sorted ascendingly by their date

And I see the total number of registrations for every event



The screenshot shows the 'Manage Events' interface. On the left is a sidebar with 'Manage Events' selected. The main content area has a header 'Manage Events' and a 'Create new event' button. Below is a list of events. The first event is 'WorldSkills Conference 2019' with the date 'August 23, 2019' and '3,546 registrations'.

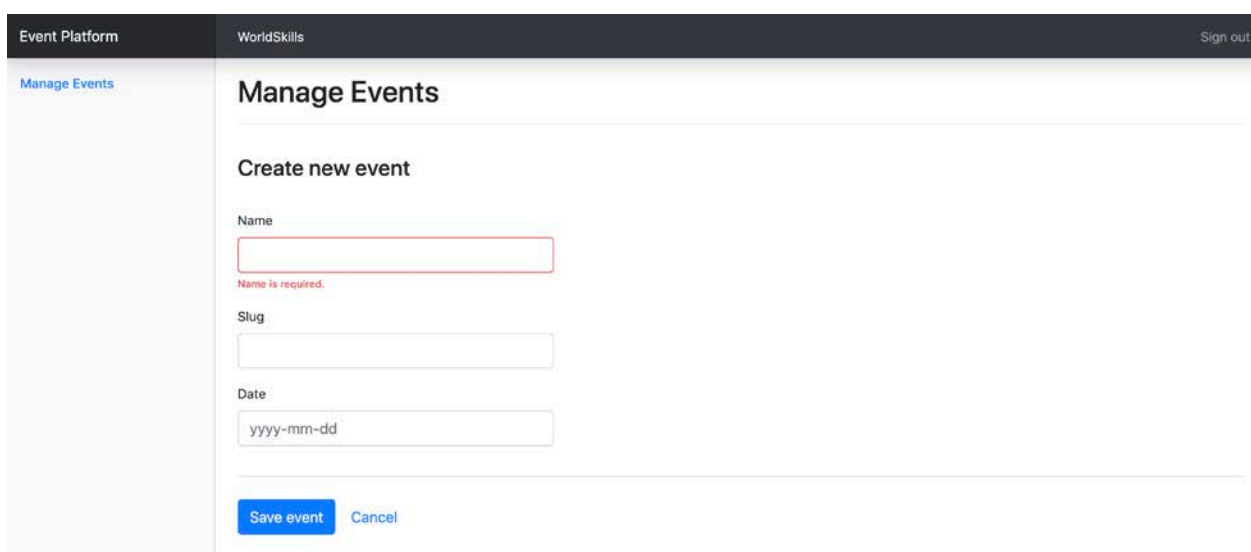
Scenario: A2b - Show form for new event

Given I am logged in

And I am on the "Manage events" page

When I press "Create new event"

Then a form to enter event information appears



The screenshot shows the 'Create new event' form. It has a header 'Create new event'. Below are three input fields: 'Name' (with a red border and a message 'Name is required.'), 'Slug', and 'Date' (with a placeholder 'yyyy-mm-dd'). At the bottom are two buttons: 'Save event' and 'Cancel'.

Scenario: A2c - Enter information for a new event

Given a form to enter event information is shown

When I enter info for the event (all fields required)

And I press "Save event"

Then the event is saved into the database

And I am redirected to the detail page for the newly created event

And a success message "Event successfully created" is shown

Scenario: A2d - Enter already used slug for a new event

Given a form to enter event information is shown

When I enter info for the event and use a slug which is already used by another event of the same organizer

And I press "Save event"

Then the event is not saved into the database

And an error message "Slug is already used" is shown

And my previously submitted input data is not lost

Scenario: A2e - Enter invalid slug for an event

Given a form to enter event information is shown

When I enter an invalid slug for the event

And I press "Save event"

Then the event is not saved into the database

And an error message "Slug must not be empty and only contain a-z, 0-9 and '-' is shown

Scenario: A2f - Show form for updating event

Given I am logged in

And I am on the detail page of an event

When I press "Edit event"

Then a form to update event information appears

Scenario: A2g - Update event information

Given a form to update information of an existing event

When I change the prefilled old event information to new values

And I press "Save event"

Then the updated event is saved into the database

And I am redirected to the detail page for the updated event

And a success message "Event successfully updated" is shown

Scenario: A2h - Show event detail

Given I am on the "Manage events" page

And I click on an event in the list

Then I am forwarded to the detail page on this event

And can see all basic information as well as all relations to the available rooms, channels, sessions and tickets

Event Platform

WorldSkills

Sign out

Manage Events

WORLDSKILLS CONFERENCE 2019

Overview

REPORTS

Room capacity

WorldSkills Conference 2019

August 23, 2019

Edit event

Tickets

Create new ticket

Normal

200.-

Early Bird

120.-

Available until June 1, 2019

VIP

400.-

100 tickets available

Sessions

Create new session

Time	Type	Title	Speaker	Channel
08:30 - 10:00	Talk	Keynote	An important person	Main / Room A
10:15 - 11:00	Talk	What's new in X?	Another person	Main / Room A
10:15 - 11:00	Workshop	Hands-on with Y	Another person	Side / Room C

Channels

Create new channel

Main

3 sessions, 1 room

Side

15 sessions, 2 rooms

Rooms

Create new room

Name	Capacity
Room A	1,000
Room B	100
Room C	100
Room D	250

Ticket management

For each event, organizers want to create different ticket types. Tickets can have, but don't need, one validity rule which can be one of:

- only purchasable until a given date (e.g. book before July 30, 2019 for an early bird ticket)
- only a defined amount of tickets is available (e.g. max 50 VIP tickets)

Please check existing records in the provided dump to see how validity rules should be stored.

Specification:

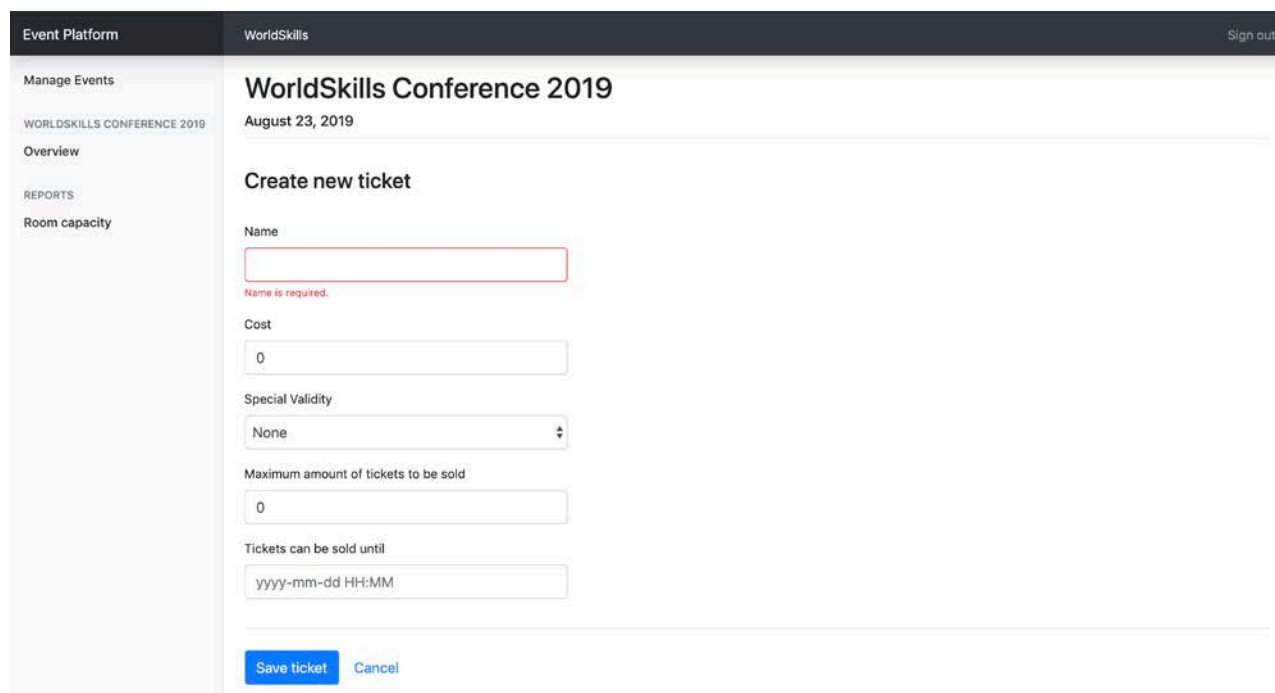
Feature: A3 - Manage tickets: As an organizer I want to manage tickets that attendees can buy in the frontend

Scenario: A3a - Show form for new ticket

Given I am on the detail page of an event

And I press "Create new ticket"

Then a form to enter ticket information appears



The screenshot shows the 'WorldSkills Conference 2019' event page. On the left is a sidebar with navigation links: 'Manage Events', 'WORLDSKILLS CONFERENCE 2019', 'Overview', 'REPORTS', and 'Room capacity'. The main content area is titled 'WorldSkills Conference 2019' with the date 'August 23, 2019'. Below this is a 'Create new ticket' form. The form contains the following fields: 'Name' (with a red border and a red error message 'Name is required.'), 'Cost' (with a value of '0'), 'Special Validity' (a dropdown menu currently set to 'None'), 'Maximum amount of tickets to be sold' (with a value of '0'), and 'Tickets can be sold until' (a date and time picker showing 'yyyy-mm-dd HH:MM'). At the bottom of the form are two buttons: 'Save ticket' (in blue) and 'Cancel' (in light blue).

Scenario: A3b - Enter information for a new ticket

Given a form to enter session information is shown

When I enter information for the ticket (name and cost field are required, also any field that is related to the selected special validity rule)

And I press "Save ticket"

Then the ticket is saved into the database

And I am redirected to the detail page of the event it is assigned to

And a success message "Ticket successfully created" is shown

Session management

For every event, organizers want to be able to list the current sessions and have the possibility to create new ones. Sessions are always assigned to a room which can not have another session during that time. Sessions are one of these two types:

- talk: A speaker presents a talk. Talks are included in the conference ticket and do not cost extra.
- workshop: A speaker holds a hand-on workshop with its attendees. Workshops can cost extra.

Specification:

Feature: A4 - Manage sessions: As an organizer I want to manage sessions that attendees can sign up for

Scenario: A4a - Show form for new session

Given I am on the detail page of an event

And I press "Create new session"

Then a form to enter session information appears

Event Platform	WorldSkills	Sign out
Manage Events WORLDSKILLS CONFERENCE 2019 Overview REPORTS Room capacity	<h2>WorldSkills Conference 2019</h2> <p>August 23, 2019</p> <h3>Create new session</h3> <p>Type <input type="text" value="Talk"/></p> <p>Title <input type="text"/> <small>Title is required.</small></p> <p>Speaker <input type="text"/></p> <p>Room <input type="text" value="Room A / Main"/></p> <p>Cost <input type="text" value="0"/></p> <p>Start <input type="text" value="yyyy-mm-dd HH:MM"/></p> <p>End <input type="text" value="yyyy-mm-dd HH:MM"/></p> <p>Description <input type="text"/></p> <p> <input type="button" value="Save session"/> <input type="button" value="Cancel"/> </p>	

Scenario: A4b - Enter information for a new session

Given a form to enter session information is shown

When I enter information for the session (all fields, except cost, required)

And I press "Save session"

Then the session is saved into the database

And I am redirected to the detail page of the event it is assigned to

And a success message "Session successfully created" is shown

Scenario: A4c - Enter invalid information for a new session - room booked

Given a form to enter session information is shown

When I choose a room which already has another session at the same time

And I press "save session"

Then the session is not saved into the database

And an error message "Room already booked during this time" is shown

And my previously submitted input data is not lost

Scenario: A4d - Show form for updating session

Given I am on the detail page of an event

And I click on the name of a session

Then a form to update session information appears

Scenario: A4e - Update session information

Given a form to update information of an existing session

When I change the prefilled old event information to new values

And I press "Save session"

Then the updated session is saved into the database

And I am redirected to the detail page of the event it is assigned to

And a success message "Session successfully updated" is shown

Channel management

A channel is used to group multiple sessions by a topic, for example management or user experience. Channels are assigned to exactly one event.

Specification:

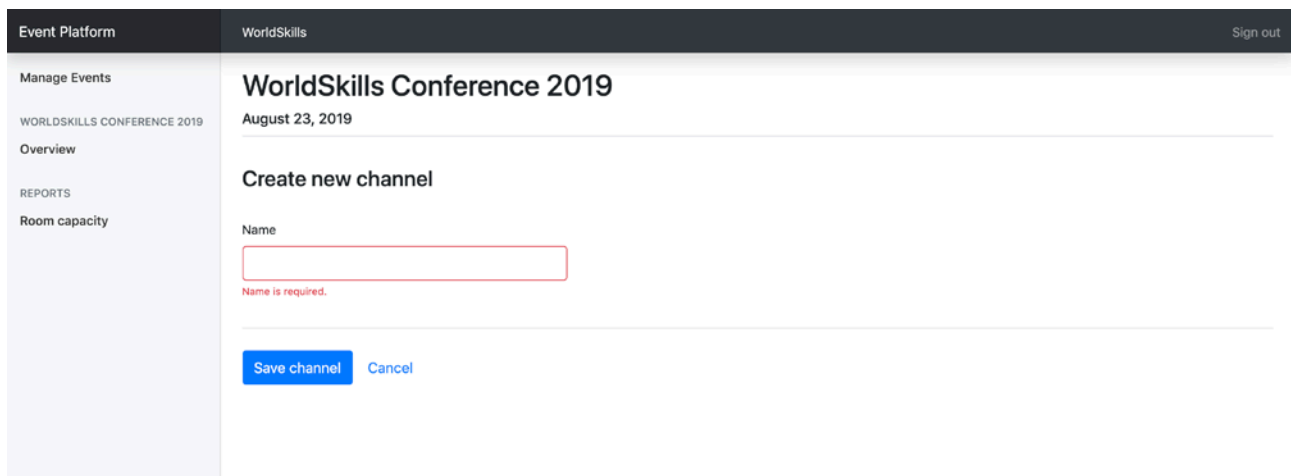
Feature: A5 - Manage channels: As an organizer I want to manage channels of an event so sessions can get assigned to them

Scenario: A5a - Show form for new channel

Given I am on the detail page of an event

And I press "Create new channel"

Then a form to enter channel information appears



The screenshot shows the 'Event Platform' interface for the 'WorldSkills Conference 2019' on 'August 23, 2019'. On the left is a sidebar with navigation links: 'Manage Events', 'WORLDSKILLS CONFERENCE 2019', 'Overview', 'REPORTS', and 'Room capacity'. The main content area is titled 'WorldSkills Conference 2019' and 'August 23, 2019'. Below this is a section titled 'Create new channel'. It contains a 'Name' label and a text input field. Below the input field is a red error message: 'Name is required.' At the bottom of the form are two buttons: 'Save channel' (in blue) and 'Cancel' (in light blue).

Scenario: A5b - Enter information for a new channel

Given a form to enter channel information is shown

When I enter information for the channel (all fields required)

And I press "Save channel"

Then the channel is saved into the database

And I am redirected to the detail page of the event it is assigned to

And a success message "Channel successfully created" is shown

Room management

A room is assigned to exactly one channel and can have exactly one session at a time.

Specification:

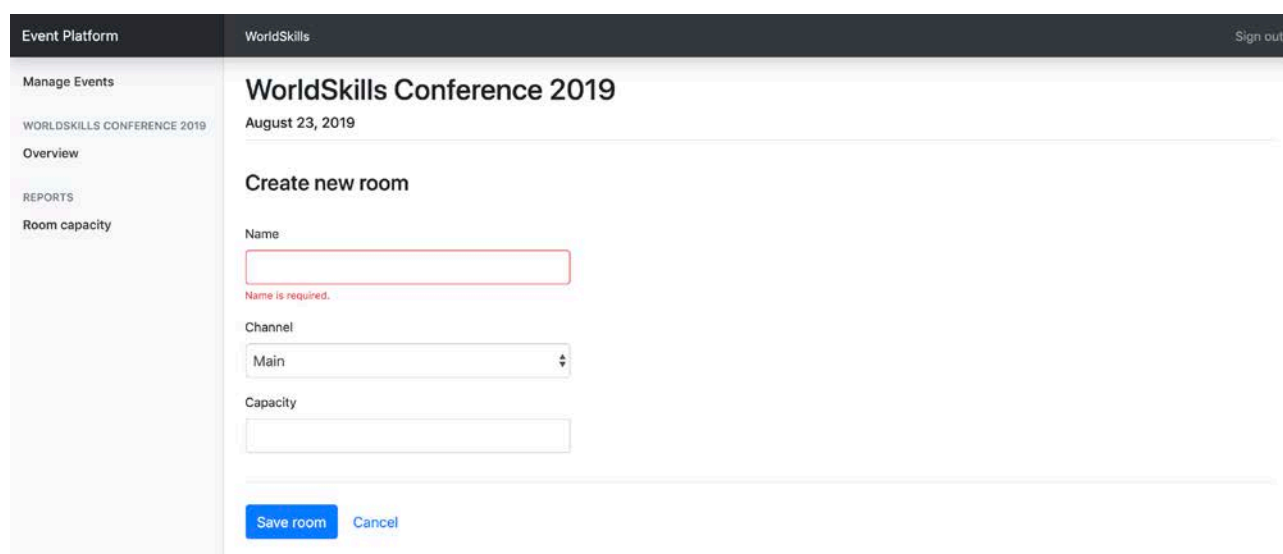
Feature: A6 - Manage rooms: As an organizer I want to manage rooms which I can later assign to sessions

Scenario: A6a - Show form for new room

Given I am on the detail page of an event

And I press "Create new room"

Then a form to enter room information appears



The screenshot shows the 'Event Platform' interface for 'WorldSkills Conference 2019' on 'August 23, 2019'. The left sidebar contains navigation links: 'Manage Events', 'WORLD SKILLS CONFERENCE 2019', 'Overview', 'REPORTS', and 'Room capacity'. The main content area is titled 'WorldSkills Conference 2019' and 'August 23, 2019'. Below this is a section titled 'Create new room'. It contains three input fields: 'Name' (with a red border and a message 'Name is required.'), 'Channel' (a dropdown menu with 'Main' selected), and 'Capacity' (an empty text box). At the bottom of the form are two buttons: 'Save room' (blue) and 'Cancel' (light blue).

Scenario: A6b - Enter information for a new room

Given a form to enter room information is shown

When I enter information for the room (all fields required)

And I press "Save room"

Then the room is saved into the database

And I am redirected to the detail page of the event it is assigned to

And a success message "Room successfully created" is shown

Room capacity

Organizers want to see an overview of all sessions with information about the room capacity and number of attendees. With this information, they can see if a room is reaching its full capacity.

Please note that for all sessions of type talk, attendees don't need a separate ticket. For sessions of type workshop, they need a separate ticket.

Specification:

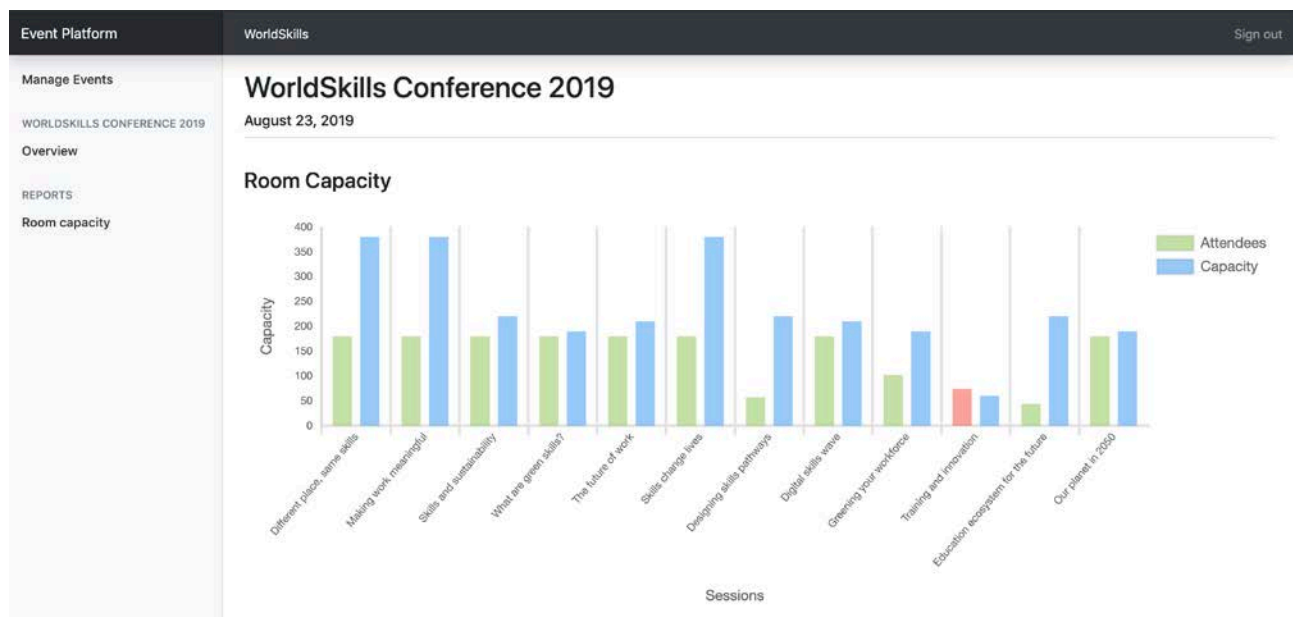
Feature: A7 – Room overview: As an organizer I want to see an overview of the sessions with the room capacity and number of attendees

Scenario: A7a – Show room overview

Given I am on the detail page of an event

When I press the link "Room capacity" in the sidebar

Then I see a bar chart displaying sessions along the horizontal axis, while the y-axis is showing the room capacity superimposed with the registered amount of attendees



Scenario: A7b – Room capacity exceeded

Given I am on the "Room capacity" page

When I have a room which exceeds its capacity

Then I see the visitors bar highlighted with a red color instead of green

Attendee API

Create a RESTful API (http://<hostname>/XX_JS-PHP_A/api/v1/...) to

- login and logout as an attendee
- read all upcoming events
- see the agenda for an event
- sign up and purchase a ticket for an event

PHPUnit tests

Tests for the PHP API are distributed with the mediafiles. Please see the following instructions to set them up and configure on your own server.

Setup:

1. Upload the API tests to your server
2. Configure the `.env` file of the API tests and provide correct database credentials and the URL where your API is available.

Running the tests:

Execute the following command on the server inside the directory of the api tests:

```
composer test
```

To only run a single test, you can use the following command for example:

```
composer test -- --filter B2EventDetailTest
```

Additional information for the REST-API tests/specification: the body contains some static example data, adjust it to fit the requirements. Placeholder parameters in the URL are marked with curly braces: {slug}

The tests are referenced by its function name [-> `test<scenario-id><text>`].

The order of properties in objects does not matter, but the order of items in an array does.

Events overview

When an attendee lands on the index page of the application, he will get a list with all upcoming events across all organizers. Events should be ordered ascendingly by their date.

Specification:

Feature: B1 – Read events: As a visitor I want to read the public data of all upcoming events so that I can display them in a list

Scenario: Bla – Read all upcoming events

Request

URL: `/api/v1/events`

Method: `GET`

Header:

Body: -

Response

If success [-> `testBlaGetEvents`]

Header: Response code: 200

```
Body: {"events": [{"id": 1, "name": "someText", "slug": "some-text",
"date": "2019-08-15", "organizer": {"id": 1, "name": "someText", "slug":
"some-text"}}]}
```

Event detail

When an event is selected, all information of the selected event should be returned in the same endpoint.

If a session has no cost or a ticket no description, NULL will be used.

The description of a ticket is generated as following:

- NULL if no special validity rule set
- "Available until {month} {day}, {year}" if special validity rule "date" is used
(example: "Available until September 1, 2019")
- "X tickets available" if special validity rule "amount" is used
(example: "30 tickets available" where 30 is the total amount of tickets and not the remaining ones)

Specification:

Feature: B2 - Read event: As a visitor I want to read all information about an event

Scenario: B2a - Read all detail information of a single event

Request

URL: /api/v1/organizers/{organizer-slug}/events/{event-slug}

Method: GET

Header:

Body: -

Response

If success [-> testB2aGetEventDetail | testB2aTicketValidity]

Header: Response code: 200

```
Body: {"id": 1, "name": "someText", "slug": "some-text", "date":
"2019-08-15", "channels": [{"id": 1, "name": "someText", "rooms": [{"id": 1,
"name": "someText", "sessions": [{"id": 1, "title": "someText",
"description": "someText", "speaker": "someText", "start": "2019-08-15
10:00:00", "end": "2019-08-15 10:45:00", "type": "workshop", "cost":
50|null}]}]}], "tickets": [{"id": 1, "name": "someText", "description":
"Available until July 7, 2019|null", "cost": 199.99, "available": true}]}
```

If event does not exist or was not created by the organizer [-> testB2aGetInvalidEvent]

Header: Response code: 404

Body: {"message": "Event not found"}

If organizer does not exist [-> testB2aGetInvalidOrganizer]

Header: Response code: 404

Body: {"message": "Organizer not found"}

Attendee login and logout

After signup (which is not within the scope of this project), attendees get assigned a random 6 characters long registration code which they need when entering an event. To avoid requiring yet another password, this registration code is also used for logging the user in on the website, together with his lastname as his login credentials. The registration code is shared across all events and is *not* only valid for a single event.

For testing purposes, the registration code and last name can be seen in the database dump.

On a successful login, data of the loggedin attendee gets returned together with a token which can be submitted with subsequent requests. For simplicity, the token is the username md5-hashed and is valid until the logout endpoint is called.

Specification:

Feature: B3 - Login and Logout as attendee: As a visitor I want to login/logout so that I can attend an event and see my registrations

Scenario: B3a - Attendee login

Request

URL: /api/v1/login

Method: POST

Header:

Body: {"lastname": "someText", "registration_code": "someText"}

Response

If success [-> *testB3aCorrectLogin/SameLastname/SameRegistrationCode*]

Header: Response code: 200

Body: {"firstname": "someText", "lastname": "someText", "username": "someText", "email": "someText", "token": "AUTHORIZATION_TOKEN"} (md5 hashed username, valid until logout)

If user login info not correct [->

testB3aInvalidLastname/RegistrationCode/Request]

Header: Response code: 401

Body: {"message": "Invalid login"}

Scenario: B3b - Attendee logout

Request

URL: /api/v1/logout?token={AUTHORIZATION_TOKEN}

Method: POST

Header:

Body: -

Response

If success [-> *testB3bLogout*]

Header: Response code: 200

Body: {"message": "Logout success"}

If invalid token [-> *testB3bInvalidToken / testB3bAlreadyLoggedIn*]

Header: Response code: 401

Body: {"message": "Invalid token"}

Event registration

Users will register and buy a ticket with this endpoint. The different ticket validity rules (date or amount) should be validated. Registrations should be sorted ascendingly by their id.

Specification:

Feature: B4 - Event registration: As a visitor I want to register for an event

Scenario: B4a - New event registration

Request

URL: /api/v1/organizers/{organizer-slug}/events/{event-slug}/registration?token={AUTHORIZATION_TOKEN}
Method: POST
Header:
Body: {"ticket_id": 1, "session_ids": [1, 2, 3]} (session_ids is optional)

Response

If success [-> *testB4aCorrectRegistration*]
Header: Response code: 200
Body: {"message": "Registration successful"}
If user is not logged in or token is invalid [-> *testB4aLoggedOut*]
Header: Response code: 401
Body: {"message": "User not logged in"}
If user already registered for this event [-> *testB4aAlreadyRegistered*]
Header: Response code: 401
Body: {"message": "User already registered"}
If ticket is not available anymore [-> *testB4aInvalidTicket*]
Header: Response code: 401
Body: {"message": "Ticket is no longer available"}

Scenario: B4b - Get registrations of a user

Request

URL: /api/v1/registrations?token={AUTHORIZATION_TOKEN}
Method: GET
Header:
Body: -

Response

If success [-> *testB4bGetRegistrations* | *testB4bNewRegistration*]
Header: Response code: 200
Body: {"registrations": [{"event": {"id": 1, "name": "someText", "slug": "some-text", "date": "2019-08-15", "organizer": {"id": 1, "name": "someText", "slug": "some-text"}}, "session_ids": [1, 2, 3]}]}
If user is not logged in or token is invalid [-> *testB4bLoggedOut*]
Header: Response code: 401
Body: {"message": "User not logged in"}

Phase two – JavaScript

API Setup

The API needed for this part will be **provided after lunch** and can be set up with the following instructions. In case you want to start with part two before lunch, you have to use your own API from part one.

Automated setup:

Connect to your server through SSH and execute the following command (in your home directory):

```
setup-api
```

This will automatically set up the API and database on your server. After completion, the API is available at <http://XX.skill17.com/day2-afternoon/> (XX is your workstation number).

The API is implemented according to the specification of the Attendees API in part one.

Manual setup:

The manual setup should only be done in case the automatic setup fails.

1. Download the API from http://wsi.skill17.com/day2_afternoon_api.zip (only available in the afternoon)
2. Upload it to your server and unzip it into the folder `/home/competitor/www/day2-afternoon` (ensure that the unzipped files are directly located in the folder `/home/competitor/www/day2-afternoon` and not in another subfolder)
3. Create a new database called `day2_afternoon` and import the provided dump from part one
4. The API is now available at <http://XX.skill17.com/day2-afternoon/> (XX is your workstation number)

Introduction

Using the RESTful API (all data coming/going into database), create a frontend for attendees to browse information about the events, login/logout and register for an event. During registration, it should be possible to add additional workshops they want to attend.

The RESTful API supports POST requests with JSON or FormData bodies, but the correct request headers need to be set accordingly to the used body type.

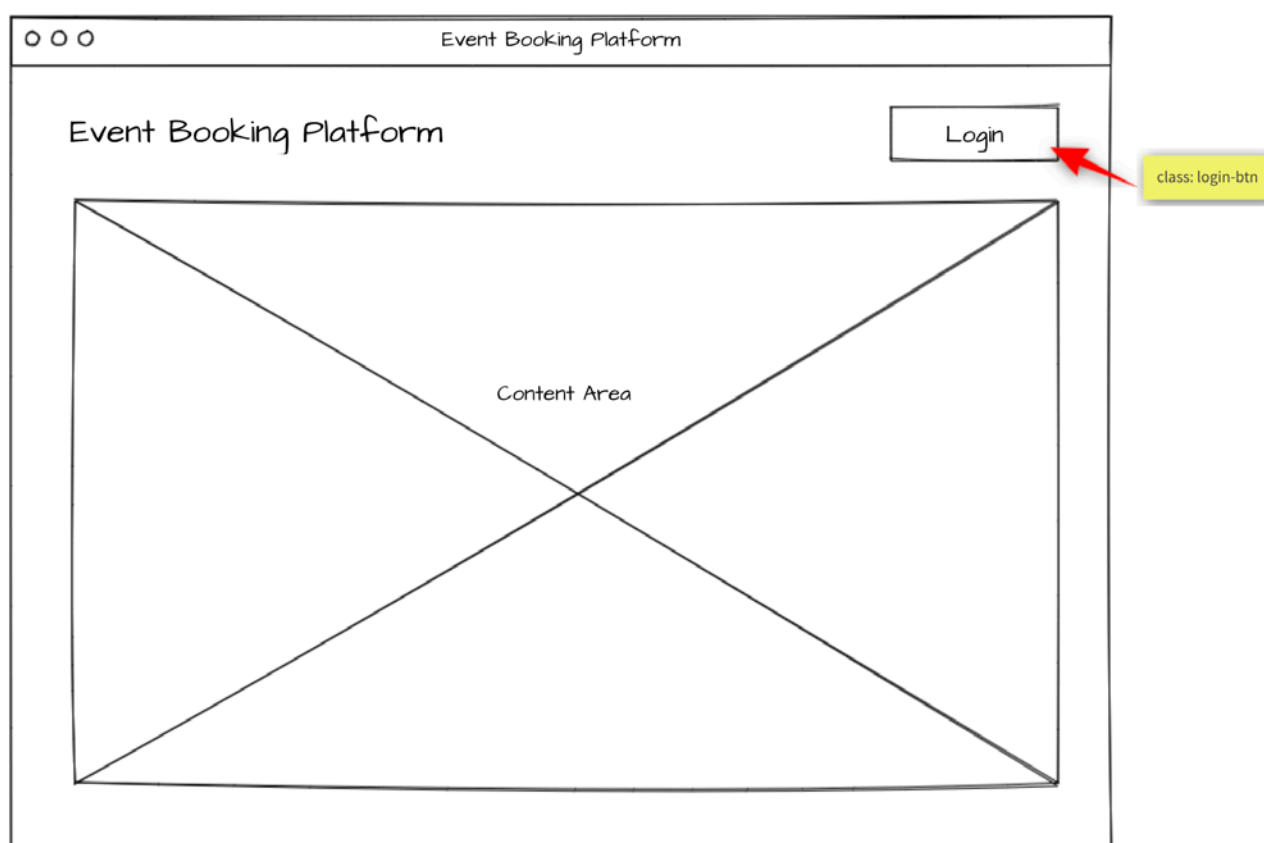
Interaction should show error/feedback messages based on response from the API where appropriate. Also, an error page should be shown when accessing an invalid URL.

Browsing all data is public and no user is required. But as soon as they want to register for an event, they need to be signed in.

Refreshing a page should always display the same page and it must be possible to share a page by URL so that all state is kept in the URL.

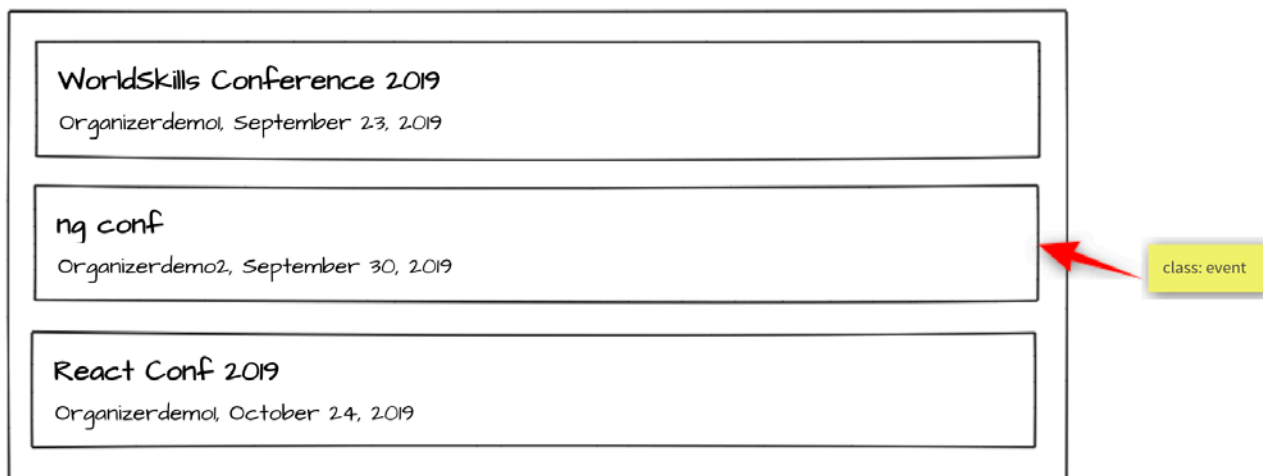
The layout should follow mockups exactly. The position of elements in design should be same as mockups.

The yellow bubbles in the mockups show classes and ids which have to be used for those elements to allow running automated tests. The yellow bubbles are not part of the design and do not have to be displayed in the browser.



Events list

When accessing the index page, all upcoming events are displayed with their organizer name and the date it takes place. By selecting an event, the event agenda will be shown.



Specification:

Feature C1 - List upcoming events: As an attendee, I want to see what events I can attend.

Scenario: C1a - List upcoming events

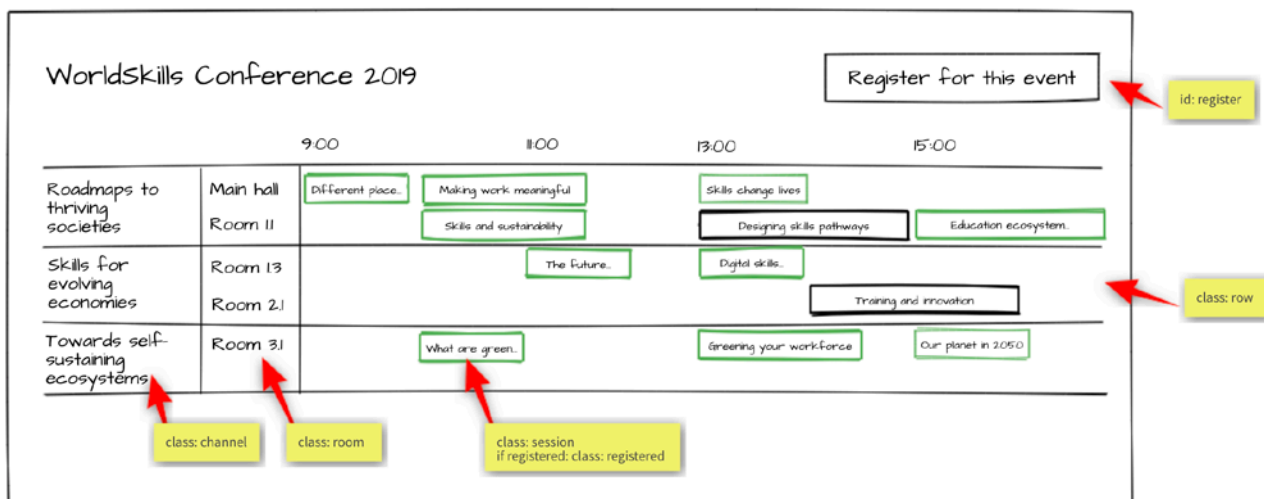
Given I am on the home page

Then I see a chronologically ordered list of upcoming events across all organizers

And I can click on an event

Event agenda

The agenda of the selected event is shown. If the user is logged in and is already registered for this event, all sessions with type talk are highlighted with a green border since they are automatically included in every event registration. Additionally, if the user signed up for addition sessions of type workshop, they will also be highlighted with a green border.



Specification:

Feature C2 - Display event agenda: As an attendee, I want to see what sessions are within an event.

Scenario: C2a - Display channels

Given I am on an event detail page

Then I see the channels of the event forming horizontal swimlanes

And the channels contain the rooms as "inside"-swimlanes

Scenario: C2b - Display talk and workshop sessions

Given I am on an event detail page

Then I see the timeline containing "9:00", "11:00", "13:00", "15:00"

And I see the sessions in the correct room swimlane starting and ending at the correct time

Scenario: C2c - Display registered sessions

Given I am on an event detail page

And I am logged in

And I am already registered for this event

Then I see all sessions of type talk highlighted with a green border

And I see all sessions of type workshop which I have signed up for highlighted with a green border

Session detail

When clicking on a session within the agenda, a detail page showing all information about this session will be shown.

Education ecosystem for the future - Workshop

In our fast-changing world, traditional forms of education are becoming outdated. We need to redesign our education systems. What are approaches to learning work in this new world?

Speaker: Pavel Luksha

Start: 15:00

End: 17:00

Cost: 30.-

Specification:

Feature C3 - Display session details: As an attendee, I want to inform myself about the details of a session.

Scenario: C3a - Display session details

Given I am on an event detail page

When I click on a session

Then I see a detail page of the session containing title, description, speaker, start and end time, type and the cost (if it is set)

Event registration

After clicking on "register for this event", the user either sees the event registration form if he is logged in or gets redirected to the login form if he is not logged in.

On the event registration form, he can select the ticket type he wants. Unavailable tickets are greyed out and can't be selected. The API is responsible for this information and the validation, so it does not have to be handled in the frontend.

The user can also select additional sessions of type workshop he wants to attend. If a session has an additional cost, it will show up in the cost view and add to the total cost.

After clicking on "Purchase", the user will be registered for this event and the sessions he has selected and gets redirected back to the event detail page.

WorldSkills Conference 2019

☒ Normal Ticket 210.-
☐ Early Bird unavailable 110.-
☐ Hotel Package 300.-
50 tickets available

Select additional workshops you want to book:

☐ Designing skills path
☒ Education ecosyste
☐ Training and innovat

Event ticket: 210.-
Additional workshops: 30.-
Total: 240.-

Purchase

class: ticket
class: workshop
id: event-cost
id: additional-cost
id: total-cost
id: purchase

Specification:

Feature: C4 - Register for event: As an attendee, I want to be able to register for an event and pay the admission fee.

Scenario: C4a - Enter registration mode

Given I am on an event detail page
And I am logged in
Then I can click on a button to enter registration mode
And I see the ticket types and their cost
And I see a checkbox list of sessions of type workshop
And I see a cost view summary
And I see a disabled "Purchase" button

Scenario: C4b - Select ticket success

Given I am in registration mode of an event
When I select a ticket that is available
Then the ticket cost is displayed in the cost view summary
And the "Purchase" button is enabled

Scenario: C4c - Select ticket fail

Given I am in registration mode of an event
When I select a ticket that is not available
Then the ticket is not selected
And the "Purchase" button is not enabled

Scenario: C4d - Select workshop session

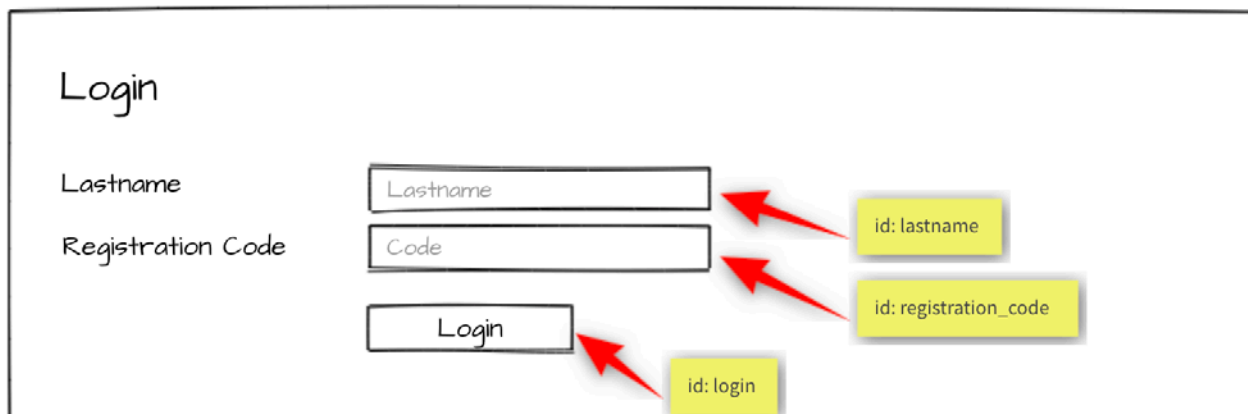
Given I am in registration mode in an event
When I tick a workshop session
Then the session receives a tick
And the price of the workshop is displayed under "Additional workshops" in
in the cost view
And the price of the session is added to the total cost

Scenario: C4e - Purchase

Given I am in registration mode in an event
When I select a ticket and I press the "Purchase" button
Then the registration is saved
And I get redirected to the event agenda
And a message appears "Registration successful"

Attendee login

When registering for an event, a login is required. After a successful login, the login button in the navigation should no longer be visible. Instead, the username of the currently logged in user should be displayed and a button to log out.



The diagram shows a login form titled "Login". It contains three input fields: "Lastname", "Registration Code", and a "Login" button. To the right of each input field is a yellow box containing an ID value. Red arrows point from each yellow box to its corresponding input field. The yellow boxes are labeled "id: lastname", "id: registration_code", and "id: login".

Specification:

Feature: C5 - Log in and log out as an attendee: As an attendee, I would like to log in and log out to the event platform so that I can access my registrations and register for new events.

Scenario: C5a - Attendee login ok

Given I am on the "login" page

When I enter the specified lastname and registration code for an attendee

And press the "Login" button

Then I'm forwarded to the previous page

And username and "Logout" button become visible in the header

Scenario: C5b - Attendee login fail

Given I am on the "login" page

When I enter a wrong lastname and registration code for an attendee

And press the "Login" button

Then I stay on the "login" page

And I see an error message "Lastname or registration code not correct"

Scenario: C5c - Attendee logout

Given I am logged in

When I press the "Logout" button

Then I am forwarded to the "login" page

And username and logout button disappear

And I cannot reach the previous page by going back

Notes

- When you are done with the first phase resp. after lunch you may continue with the second phase.
- If you are done with part one before the lunch break, you can start with part two by using your own API and exchange it at the start of the afternoon with the provided API.
- Competitors must use one of the server-side and client-side frameworks/libraries that are provided.
- The provided template for the admin interface should be used. It can be enhanced to get better functionality for your site. The visual design of the *admin interface* will *not* be assessed.
- The design of the attendee interface can be based on the template of the admin interface. The visual design of the attendee interface *will* be assessed: a simple, clean and attractive design is expected, including choice of colors, typography and spacing.
The design should be optimized for a minium window width of 1024px.
- For the admin interface it is important to use the selectors (class, id) as provided in the template. For the attendee interface it is important to use the selectors (class, id) as specified in the screen mockup (yellow bubbles).
- The specified database tables need to be implemented as provided. More tables may be added if needed. The given tables must not be changed. Provide a final SQL-dump and ERD screen as specified below.
- The API implementation should fulfill all requirements as stated in the description. All prefixes, RESTful-URLs and HTTP-Methods from the given API links must be implemented correctly and cannot be changed.
- Changes made in the data need to be propagated between frontend and backend in both directions.
- Update the first two organizers in the provided dump so it is possible to login to the system during assessment:
 - Organizer accounts:
 - Email: demo1@worldskills.org Password: demopass1
 - Email: demo2@worldskills.org Password: demopass2
- For the room capacity bar chart, you can use one of the provided JS-libraries (Charts.js, D3.js)

Submission Instructions to the Competitor

Morning: Files to be collected after the first phase on the server:

- Web service (http://<hostname>/XX_JS-PHP_A/api/v1/..)
- Admin area (http://<hostname>/XX_JS-PHP_A/) - initially shows login page
- Folder inside webroot (XX_JS-PHP_A/)
- ERD screenshot named "XX_ERD.png" in "db-dump" folder inside of XX_JS-PHP_A
- Database dump named "XX_database.sql" in "db-dump" folder inside of XX_JS-PHP_A

Afternoon: Files to be collected after the second phase on the server:

- Front-end website (http://<hostname>/XX_JS-PHP_B/)
- Folder inside webroot (XX_JS-PHP_B/) - put all development files into XX_JS-PHP_B_dev.zip and leave only files for production-use in this directory.
- If the front-end root (http://<hostname>/XX_JS-PHP_B/) does not open the main page directly, have it open an index.html with a link to the main page for marking your front-end.

XX is your country code

The Web Service / REST-API will be assessed with the **automated test-framework** provided. Additional data may be used to test dynamic behavior.

It is recommended that you run the automated tests for the REST-API after implementing it locally or at least after uploading it to the server. Instructions to setup and use the tests for the API: see README.md in tests_api-php.

Manual assessment of other parts will be done using **Google Chrome**.

Equipment, machinery, installations, and materials required

It is expected that all Test Projects can be accomplished by Competitors based on the equipment and materials specified in the Infrastructure List*.

If any equipment or material is required that is not listed in the Infrastructure List, then it has to be listed here (in the table below - column headings may vary from skill to skill.)

WorldSkills International needs to approve any additional infrastructure requirements that are not listed in the Infrastructure List. Column headings may vary from skill to skill.

* (Definition: the Infrastructure List is the equipment, machinery, installations and materials supplied by the host country – it does not include tools and material to be supplied by Competitors and/or Experts)

ITEM	DESCRIPTION
Media Files	Contains: - DB dump - Template files for the admin-gui - Charts.js & D3.js incl. Documentation - Files for testing REST-API

Marking Scheme

SECTION	CRITERION	JUDG. MARKS	MEAS. MARKS	TOTAL
C1	Database	4.75 <u>3.85</u>	4.53	3.25 <u>6.85</u>
C2	Admin Backend	0	3.25 <u>6.5</u>	3.25 <u>6.5</u>
C3	Ticket and Session	0	4.53	4.53
C4	Organizer	42.35	2.25 <u>4.5</u>	3.25 <u>6.85</u>
C5	REST API	4.53 <u>3.35</u>	2.25 <u>4.5</u>	3.75 <u>7.85</u>
D1	Events list and agenda	0	3.75 <u>7.5</u>	3.75 <u>7.5</u>
D2	Session and Events	0	36	36
D3	General	42.35	36	48.35
D4	JS Quality	2.55 <u>3.35</u>	0	2.55 <u>3.35</u>
D5	Front end design	36.35	0	36.35
TOTAL		40.75 <u>23.6</u>	20.54 <u>1</u>	31.25 <u>64.6</u>