

Test Project

Web Technologies

Vaccination Campaign Management - BackEnd

Submitted by:

Trần Đức Doanh
Lương Văn Hiếu
Khúc Thị Ngọc Hà
Trần Việt Khánh
Nguyễn Ngọc Anh
Kim Thị Thương
Kiều Thành Chung

Contents

Contents	2
Introduction	3
Description of project and tasks	3
Module A – PHP	4
Database design.....	4
Organizer interface	4
<i>Organizer login and logout.....</i>	<i>5</i>
<i>Campaign management</i>	<i>6</i>
<i>Ticket management.....</i>	<i>8</i>
<i>Session management.....</i>	<i>9</i>
<i>Area management.....</i>	<i>11</i>
<i>Place management.....</i>	<i>12</i>
<i>Place capacity.....</i>	<i>12</i>
<i>Citizen API.....</i>	<i>14</i>
<i>Overview of Campaigns.....</i>	<i>14</i>
<i>Campaign detail</i>	<i>15</i>
<i>Citizen login and logout.....</i>	<i>15</i>
<i>Campaign registration.....</i>	<i>16</i>
Notes.....	17
Submission Instructions to the Competitor	18
Equipment, machinery, installations, and materials required.....	19
Marking Scheme	19

Introduction

In 2021, the second wave of Covid-19 has hit Vietnam and the world. In order to reach community immunity, the National Covid-19 Vaccination Campaign had to be implemented as soon as possible. The government, who is responsible for organizing the Covid-19 Vaccination Campaign, needs a new web application helping them to manage the campaign and allowing the unvaccinated and vaccinated citizens to sign up and see 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:

- Module A (morning, 3.5 hrs) for building the backend parts (admin interface and REST-API) using PHP (framework) and database
- Module B (afternoon, 3.5 hrs) for building the frontend parts (attendee interface) using HTML/CSS/JS (framework) and the REST-API provided for phase two.

Module A – PHP

Database design

- Use your database (db_dump) 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)
 - Campaign ratings
 - Citizen should be able to comment and rate campaign they have participated
 - 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 campaign rating, citizen should be able to rate sessions they have participated
 - 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 campaigns, sessions, areas, places and tickets (create, read, update)
- create a bar chart to see the occupancy rates of all places
- 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 campaigns of other organizers or update campaigns 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@vietnamskills.org Password: demopass1
- Email: demo2@vietnamskills.org Password: demopass2

The covid 19 vaccination campaign

Please sign in

Email
Password

Sign in

Specification:

Feature: D1 – Log in and log out as an organizer: As an organizer, I would like to log in and log out to the campaign platform so that I can access the functionality of the platform.

Scenario: D1a – 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 campaigns" becomes visible

Scenario: D1b – 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 a participant 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: D1c – 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

Campaign management

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

Specification:

Feature: D2 - Manage campaigns: As an organizer I want to manage campaigns so participants can find campaigns

Scenario: D2a - List all campaigns

Given I am logged in

And I am on the "Manage campaigns" page

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

And I see the total number of registrations for every campaign

CVC Platform	{insert organization name}	Sign out
Manage Campaigns	<h2>Manage Campaigns</h2> <div> <p>September 12, 2021 campaign {insert campaign date}</p> <p>3,546 registrations</p> </div>	Create new campaigns

Scenario: D2b - Show form for new campaign

Given I am logged in

And I am on the "Manage campaigns" page

When I press "Create new campaign"

Then a form to enter campaign information appears

CVC Backend	{insert organization name}
Manage Campaigns	<h2>Manage Campaigns</h2> <h3>Create new campaign</h3> <p>Name</p> <input type="text"/> <p>Name is required.</p> <p>Slug</p> <input type="text"/> <p>Date</p> <input type="text" value="yyyy-mm-dd"/> <p>Save campaign Cancel</p>

Scenario: D2c - Enter information for a new campaign

Given a form to enter campaign information is shown

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

And I press "Save campaign"

Then the campaign is saved into the database

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

And a success message "Campaign successfully created" is shown

Scenario: D2d - Enter already used slug for a new campaign

Given a form to enter campaign information is shown

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

And I press "Save campaign"

Then the campaign 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: D2e - Enter invalid slug for a campaign

Given a form to enter campaign information is shown

When I enter an invalid slug for the campaign

And I press "Save campaign "

Then the campaign 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: D2f - Show form for updating campaign

Given I am logged in

And I am on the detail page of a campaign

When I press "Edit campaign"

Then a form to update campaign information appears

Scenario: D2g - Update campaign information

Given a form to update information of an existing campaign

When I change the prefilled old campaign information to new values

And I press "Save campaign"

Then the updated campaign is saved into the database

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

And a success message "Campaign successfully updated" is shown

Scenario: D2h - Show campaign detail

Given I am on the "Manage campaign" page

And I click on a campaign in the list

Then I am forwarded to the detail page on this campaign

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

Event Platform
(insert organization name)
Sign out

Manage Campaigns

(insert campaign name)

Overview

REPORTS

Place capacity

{insert campaign name}

{insert campaign date}

Edit campaign

Tickets

Create new ticket

Normal

200.-

Early Bird

120.-

Available until September 12, 2021

VIP

400.-

100 tickets available

Sessions

Create new session

Time	Type	Title	Participant	Area
08:30 - 10:00	Normal	Keynote	An important person	Main / Place A
10:15 - 11:00	Normal	What's new in X?	Another person	Main / Place A
10:15 - 11:00	Service	Hands-on with Y	Another person	Side / Place C

Areas

Create new area

Main

3 sessions, 1 place

Side

15 sessions, 2 places

Places

Create new place

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

Ticket management

For each campaign, 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 Dec 31, 2021 for an early bird ticket)
- only 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: D3 - Manage tickets: As an organizer I want to manage tickets that participants can buy in the frontend

Scenario: D3a - Show form for new ticket

Given I am on the detail page of a campaign

And I press "Create new ticket"

Then a form to enter ticket information appears

CVC Backend
(insert organization name)
Sign out

Manage Campaigns

(INSET CAMPAIGN NAME)

Overview

REPORTS

Place capacity

{insert campaign name}

{insert campaign date}

Create new ticket

Name

Name is required.

Cost

Special Validity

None

Maximum amount of tickets to be sold

Tickets can be sold until

Save ticket Cancel

Scenario: D3b – 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 campaign it is assigned to

And a success message "Ticket successfully created" is shown

Session management

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

- **Normal:** The participant will be vaccinated with standard procedures.
- **Service:** The participants will be vaccinated using extra services which allow them to access air-conditioned waiting room to be injected without queuing and use private patient room. It will cost extra.

Specification:

Feature: D4 – Manage sessions: As an organizer I want to manage sessions that participants can sign up for

Scenario: D4a – Show form for new session

Given I am on the detail page of a campaign

And I press "Create new session"

Then a form to enter session information appears

CVC Backend
(insert organization name)
Sign out

Manage Campaigns

(INsert campaign name)

Overview

REPORTS

Place capacity

{insert campaign name}

(insert campaign date)

Create new session

Type

Normal

Title

Title is required.

Participant

Place

Place A / Main

Cost

0

Start

yyyy-mm-dd HH:MM

End

yyyy-mm-dd HH:MM

Description

Save session
Cancel

Scenario: D4b – 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 campaign it is assigned to

And a success message "Session successfully created" is shown

Scenario: D4c – Enter invalid information for a new session – places booked

Given a form to enter session information is shown

When I choose a place 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 "Places already booked during this time" is shown

And my previously submitted input data is not lost

Scenario: D4d – Show form for updating session

Given I am on the detail page of a campaign

And I click on the name of a session

Then a form to update session information appears

Scenario: D4e – Update session information

Given a form to update information of an existing session

When I change the prefilled old campaign 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 campaign it is assigned to

And a success message "Session successfully updated" is shown

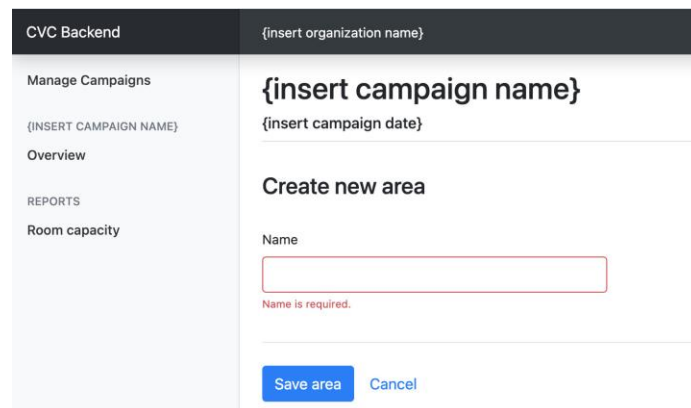
Area management

An area is used to group multiple sessions by a topic, for example management or user experience. Areas are assigned to exactly one campaign.

Specification:

Feature: D5 - Manage areas: As an organizer I want to manage areas of a campaign so sessions can get assigned to them

Scenario: D5a - Show form for new area
Given I am on the detail page of a campaign
And I press "Create new area"
Then a form to enter area information appears



Scenario: D5b - Enter information for a new area
Given a form to enter area information is shown
When I enter information for the area (all fields required)
And I press "Save area"
Then the area is saved into the database
And I am redirected to the detail page of the campaign it is assigned to
And a success message "Area successfully created" is shown

Place management

A place is assigned to exactly one area and can have exactly one session at a time.

Specification:

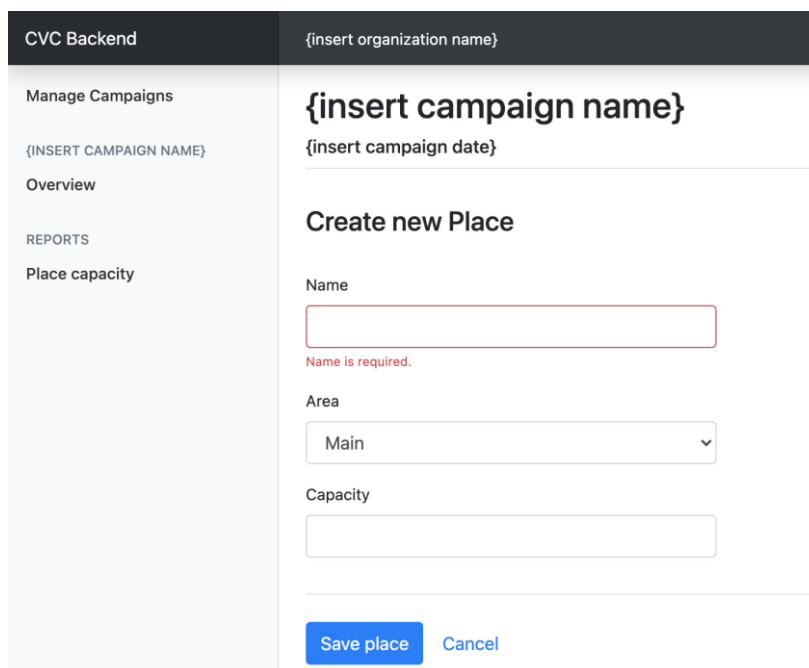
Feature: D6 – Manage places: As an organizer I want to manage places which I can later assign to sessions

Scenario: D6a – Show form for new place

Given I am on the detail page of a campaign

And I press "Create new place"

Then a form to enter place information appears



Scenario: D6b – Enter information for a new place

Given a form to enter place information is shown

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

And I press "Save place"

Then the place is saved into the database

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

And a success message "Place successfully created" is shown

Place capacity

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

Please note that for all sessions of type Nomal, participants don't need a separate ticket. For sessions of type Service, they need a separate ticket.

Specification:

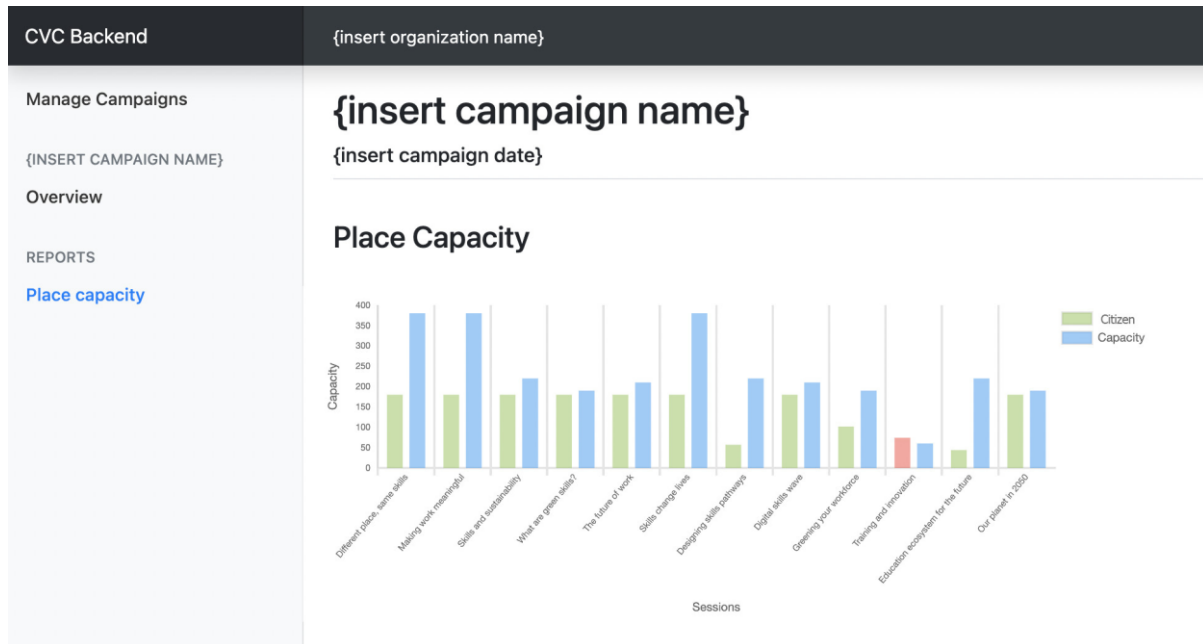
Feature: D7 – Place overview: As an organizer I want to see an overview of the sessions with the place capacity and number of participants

Scenario: D7a – Show place overview

Given I am on the detail page of a campaign

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

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



Citizen API

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

- login and logout as a citizen
- read all upcoming campaigns
- see the agenda for a campaign
- sign up and purchase a ticket for a campaign

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.

Overview of Campaigns

When a citizen lands on the index page of the application, he will get a list with all upcoming campaigns across all organizers. Campaigns should be ordered ascendingly by their date.

Specification:

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

Scenario: B1a - Read all upcoming campaigns

Request

URL: /api/v1/campaigns

Method: GET

Header:

Body: -

Response

If success [-> testB1aGetCampaigns]

Header: Response code: 200

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

Campaign detail

When a campaign is selected, all information of the selected campaign 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, 2021")
- "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 campaign: As a visitor I want to read all information about a campaign

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

Request

URL: /api/v1/organizers/{organizer-slug}/campaigns/{campaign-slug}

Method: GET

Header:

Body: -

Response

If success

Header: Response code: 200

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

If campaign does not exist or was not created by the organizer

Header: Response code: 404

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

If organizer does not exist

Header: Response code: 404

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

Citizen login and logout

After signup (which is not within the scope of this project), citizens get assigned a random 6 characters long registration code which they need when entering a campaign. 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 campaigns and is *not* only valid for a single campaign.

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

On a successful login, data of the loggedin citizens 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 citizen: As a visitor I want to login/logout so that I can attend a campaign and see my registrations

Scenario: B3a - Citizen login

Request

URL: /api/v1/login

Method: POST

Header:

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

Response

If success

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

Header: Response code: 401

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

Scenario: B3b - Citizen logout

Request

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

Method: POST

Header:

Body: -

Response

If success

Header: Response code: 200

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

If invalid token

Header: Response code: 401

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

Campaign 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 - Campaign registration: As a visitor I want to register for an campaign

Scenario: B4a - New campaign registration

Request

URL: /api/v1/organizers/{organizer-slug}/campaigns/{campaign-slug}/registration?token={AUTHORIZATION_TOKEN}

Method: POST

Header:

Body: {"ticket_id": 1, "session_ids": [1, 2, 3]} (session_ids is optional)

Response

If success

Header: Response code: 200

Body: {"message": "Registration successful"}

If user is not logged in or token is invalid

Header: Response code: 401

Body: {"message": "User not logged in"}


```
If user already registered for this campaign
  Header: Response code: 401
  Body: {"message": "User already registered"}
If ticket is not available anymore
  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

```
Header: Response code: 200
```

```
Body: {"registrations": [{"campaign": {"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

```
Header: Response code: 401
```

```
Body: {"message": "User not logged in"}
```

Notes

- Competitors must use one server-side frameworks/libraries that are provided (if no framework is used, competitor's mark will be deducted by 50% mark accordingly)
- 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 citizen 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 minimum 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.
- Update the first two organizers in the provided dump so it is possible to login to the system during assessment:
 - Organizer accounts:
 - Email: demo1@vietnamskills.org Password: demopass1
 - Email: demo2@vietnamskills.org Password: demopass2

Submission Instructions to the Competitor

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

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

XX is competitor's workstation number.

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.)

VietnamSkills 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
D1	Database	1.5	2.5	4.0
D2	Admin Backend	0	5.5	5.5
D3	Ticket and Session	0	3.5	3.5
D4	Organizer	1.0	5.5	6.5
D5	REST API	0.5	6.0	6.5
TOTAL		3.0	23.0	26.0