Unofficial

Web Development Competition 2022

TEST PROJECT

Written by:

Roberts Flaumanis,

Franz Stimpfl,

Margit Tennosaar

SKILL **17** web
technologies

Introduction

Preparing for a Skills-competitions requires a lot of work and dedication. To analyze the effectiveness of competitors' training, we are creating a time monitoring web application. This application is used for logging competitors' training sessions. Sessions are stored in a database, and charts and data tables will help visualize data.

Your task is to build **MVP for training time monitor web application** - competitors' and experts' dashboards. Use our provided prototype to follow the required layout and the functionality of the application. Application prototype

We have provided to you Vuexy Admin Dashboard Template based on Vue CLI, Vuex & Vuesax component framework. Vuexy – Vuejs, React, Angular, HTML & Laravel Admin Dashboard Template. Vuexy demo

> **Important note:**
>
> Please note that the Vuexy template is licensed, and the version you are provided is not allowed to be used anywhere other than for this competition's purposes.
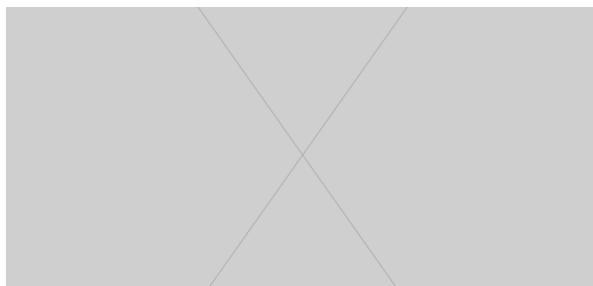
Description of the project

MVP has one type of user - competitors. All data comes from a database and should not be hard-coded.

Competitors can:

- Log in and log out
- See personal dashboard with data cards, charts and data table
- Add and modify entries

Because different users will use this web application, keep security in mind. It should not be possible to access the dashboard of other users or update the data of other users.

Login

Email

email@skill17.com

Password

****

SIGN IN

Email: **competitor1@skill17.com**   Password: **demopass1**
Email: **competitor2@skill17.com**   Password: **demopass2**

Keep the state of the art security approach to store passwords in the database in mind!

Also keep in mind to only show data, which the logged-in user is allowed to see. Changing for example an id in the URL should not allow seeing other competitors' data.

Specifications:

Feature A1 – Log in and log out as a user:

As a user, I can log in and log out of the application to access the functionality of the application.

Scenario: A1a - User login ok

> *The user is on the "login" page.*
> *The user enters the correct email and password for a user*
> *The user presses the "Sign in" button*
> *Login is successful, and the user is forwarded to the users' Dashboard*
> *And the user name is displayed in the header*
> *And "Sign in" button is replaced with the "Sign out" button in the header*

Scenario: A1b - User login fail

> *The user is on the "login" page*
> *The user enters a wrong email and password combination for the user*
> *The user presses the "Sign in" button*
> *Login is unsuccessful and the user stays on the "login" page.*
> *And the error message "Email or password not correct" is displayed.*

Scenario: A1c - User logout

> *The user is logged in*
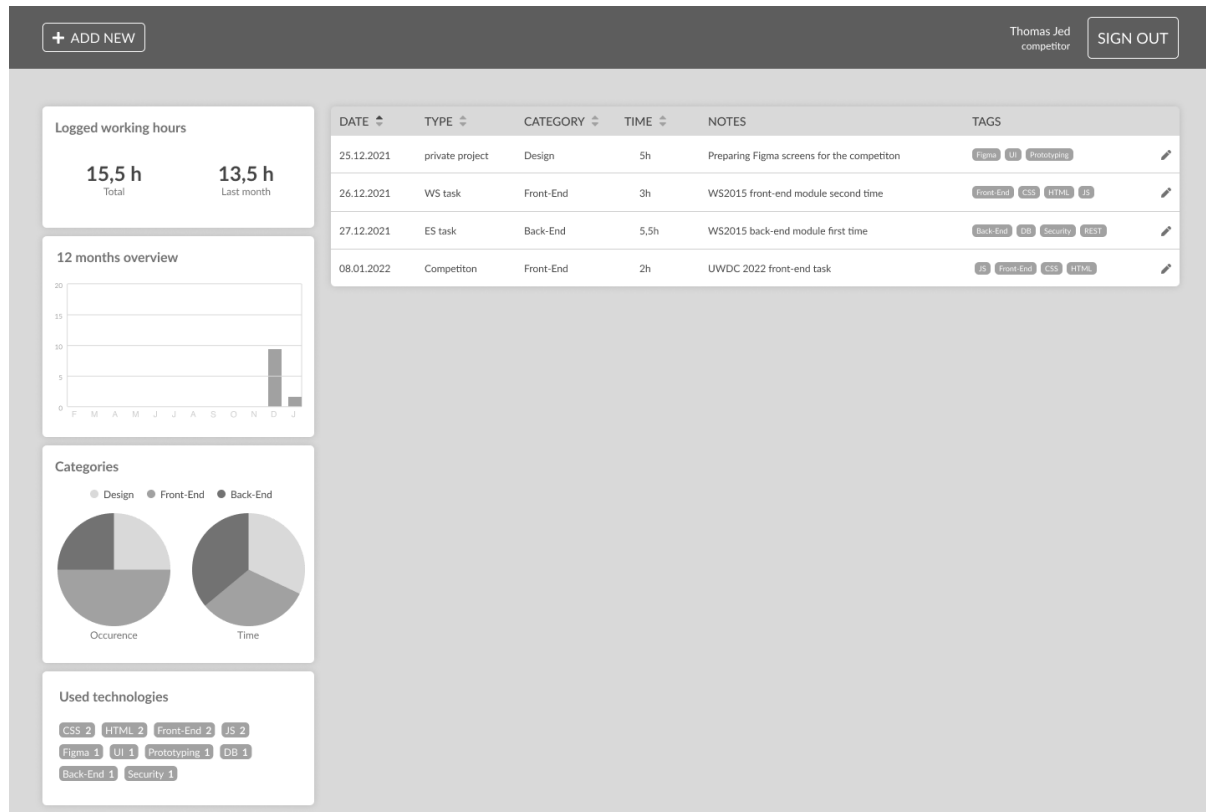> *The user presses the "Sign out" button*
> *The user is forwarded to the "login" page*
> *And user name and logout button and dashboard functionalities disappear*
> *And user cannot reach the previous page by going back*

Competitor's dashboard interface

All entered information about training sessions will be displayed on the competitor's dashboard. All data should come from a database and should not be hard-coded. Dashboard left-side data cards order is customizable by drag and drop (sortable) functionality.

Specifications:

Feature B1 - Competitor's Dashboard view:

As a competitor, I want to see my training session data so my expert and I get a better overview of my training progress.

Scenario B1a - List all training session entries

 The competitor is on the Dashboard
 Competitor see a Data Table of all training sessions sorted ascendingly by their date

Scenario B1b - Sort training session entries

 The competitor is on the Dashboard
 A data Table is shown
 The competitor presses sorting arrows in table headings (date, type, category and time)
 And Data Table training session entries are sorted accordingly

Scenario B1c - Sum of working hours data card

The competitor is on the Dashboard
A data table is shown
The competitor see "Logged working hours" data card
And this card has a sum of all entered working hours
And sum for last month hours

Scenario B1d - 12 months overview data card

The competitor is on the Dashboard
A data table is shown
The competitor see "12 months overview" data card
And this card has a bar chart of the last 12 months working hours.

Scenario B1e - Used technologies tags cloud data card

The competitor is on the Dashboard
A data table is shown
The competitor see "Used technologies" data card
And this card display all tags entered
And each tag is displayed only once
And tag label contains the number of times that tag has been used in the table
And the tag labels are sorted in descending order by most used tags.

Scenario B1f - Customizing Dashboard data cards order

The competitor is on the Dashboard
The competitor see data cards on the left side of the page
The competitor drags one of the data cards and releases the card in a different position on the left side
Then the data card switches position and order with other data cards on the left side.
And data table stays on the right side

Training sessions management

All entered information about training sessions will be saved to the database. Every competitor can see and edit only their events.

Each training session entry has:

- date *,
- type *,
- category *,
- spent time *,
- notes,
- and technology tags *.

* marks required fields for the users.

When adding training session entries, a competitor can use predefined categories, types and tags OR add new ones, which will be saved to the database. Newly added categories, types or tags should be selectable when adding the next training session.

Specification:

Feature D1 - Training sessions management:

As a competitor, I want to manage my training session entries so my expert and I can use it in my data analysis.

Scenario D1a - Enter information about a new training session

> *The competitor is on the Dashboard*
> *The competitor presses the "+ Add New" button*
> *A form to enter training session information is shown*
> *When competitor enters info about the training session (required fields)*
> *And press "Add Entry"*
> *Then the training session data is saved into the database*
> *And the new entry is displayed in the Data Table in the Dashboard*
> *And a success message "Entry added successfully" is shown*

Scenario D1b - Modify information of training session

> *The competitor is on the Dashboard*
> *The data table is shown*
> *The competitor presses the "pencil" editing button on a specific data row*
> *A form to edit training session information is shown*
> *When competitor enters/edits info about the training session (required fields)*
> *And press "Save Entry"*
> *Then the edited training session data is saved into the database*
> *And the edited entry is displayed in the Data Table in the Dashboard*
> *And a success message "Entry edited successfully" is shown*

Submission Instructions to the Competitor

Your final and built project should be stored in the default document root „/var/www/html" on the provided server and should be accessible via the provided URL. More information can be found in the readme.txt on the server.

Upload your final project with all necessary files latest on **08.01.2022 at 16:00 (HKI time - GMT+2)**.

The project must include a README.md file with installation and usage instructions. Please make sure that you include all necessary information for running your solution. Readme quality will be assessed.

Since using version control is a fundamental part of a web developer's life, we also want you to use GIT as a version control system and show us that you are capable of using GIT. The proper use of GIT does also include a meaningful usage of GIT commits. Therefore, please also provide us the link/information to your GIT repository.

Materials provided

ITEM	DESCRIPTION
Vuexy Admin Dashboard Template	Vuejs, React, Angular, HTML & Laravel Admin Dashboard Template. vuexy-admin-v8.0.0.zip 256.4MB

Marking Scheme

SECTION	MODULE	MARKS
A	Functionality	36.00
B	Layout (Client-Side)	18.00
C	Server-Side	37.00
D	Documentation	9.00
		100.00