Web Technologies
WorldSkills 2024 National Competition
HUNGARY
Round 1

Submitted by:
Skills IT

Contents

Introduction

You used to work as a freelance web developer, but now you've applied for a job as a developer at a large software development company. The company's management wants to test your skills, so as part of the recruitment process, they asked you to develop a prototype web application for the Ultra Balaton (UB) Running Festival. In addition to sharing up-to-date information about the running festival, the future web application aims to provide an easy-to-use tool for the running teams to distribute the stages.

The aim of the UB Running Fest to run around Lake Balaton individually or in a team. Teams of up to 10 people can sign up the team event. The organisers will specify the stages in advance and team members can allocate them among themselves.

The first 7 km long stage starts from Balatonfüred and goes to Aszófő, so Aszófő is the first possible place to change runners. Any number of stages can be assigned to one runner. A runner's stages can be consecutive, but a runner can also run separate stages several times.

In the prototype of the webapp, all you need to develop is an attractive landing page and a simplified version of the stage calculator.

Description of project and tasks

Your task is divided into two parts.

1. In the first part, you have to create a one-page landing page for the teams who applied in team category. On the page some basic information will be shared about the interactive stage calculator.
2. In the second part, you will develop a simplified version of the stage calculator.

How to submit your work

1. You have to share your work in a private GitHub repo as described in the README file of the test project GitHub repo (https://github.com/skillsit-hu/ws2024-s17-hu-r1), and publish the final work using the Netlify service.
2. The second part mentioned above must be made available via a relative URL `calculator` within the published site. So, for example, if your site is available at https://winniedepooh2024.netlify.app, the URL of part 2 should be https://winniedepooh2024.netlify.app/calculator.

Part 1 – Landing Page

For the landing page, you need to create a simple but attractive landing page.

- the website uses exactly the full width and height of the browser, so there's no empty space and no scrolling.
- Some form of media should also be used to attract users.
- In addition, the following information should be visible:
    - headline: ULTRABALATON
    - subtitle: Team Competition
    - Event date: 06.05.2023
- Short text describing the stage calculator. (You can find the text in the media files.)
- A button with a link to the stage calculator.
- Four social media icons (Facebook, Instagram, Pinterest, Tumblr) should be placed at the bottom of the page.
    - Icons don’t need to point to a valid URL.
    - All required files can be found in the mediafiles/social-icon-task folder.
- Feel free to add any other information and elements you find useful.
- The webpage must be responsive and support at least the following viewports:
    - Mobile: 360x640
    - Tablet: 768x1024
    - Desktop: 1920x1080

Part 2 – Route Assignment Calculator

In this part, you need to create an interactive tool to help teams allocate each part of the route.

The following parts should appear in the page:

- Team Member table
- Stage Assignement table
- Button with link to the landing page

You need to fetch the basic information about the stages from a public REST API endpoint. (If for some reason you can't do this, you can use the stages.json file in the media files as a workaround.)

The calculator tool in this prototype version does not need to be responsive, the evaluation is done on a full HD desktop (1920x1080) display.

However, in desktop view, it should be well designed, elegant, and easy to use. The look and feel should match the style of the landing page.

Team Member Table

The Team member table must be created using flexbox or CSS grid technology. The table contains the following fields:

- line number (automatically filled with line numbers 1-10)
- first name
- last name
- speed (estimated time to run one km in MM:SS format)
- total distance (calculated value, number rounded to one decimal place, sum of the distances undertaken by the runner, see later)

The table has a header row with the title of the fields and 10 rows for the data. Any field other than the line number and total distance fields can be freely edited. Initially, the table is empty.

The speed field should have the appropriate input mask (MM:SS), so when the user enters the numbers, they get formatted correctly**.** For example, if the user types 1234 the field should display 12:34.

Stage Assignment Table

The Stage Assignment table must be also created using flexbox or CSS grid technology. The table contains the following fields:

- Line number (#, automatically filled with line numbers)
- Distance (in km)
- Starting Point (name of the location of starting point)
- Arriving Point (name of the location of ending point)
- Name (name of the sponsor of the stage)
- Runner (name of the runner)
- Time (the time needed for the runner to complete the distance of the stage in MM:SS format)

All basic data except the runner and time fields are provided by the following public REST API endpoint:

https://ub2023-backend.onrender.com/api/v1/stages/

If the request is successful, the endpoint returns the following data structure in JSON format:

```
[
	{
		id: number,
		distance: number,
		startingLocation: string,
		arrivalLocation: string,
		name: string
	}
]
```

The runner field allows you to select a runner from a drop-down list. The first item in the list is always "No runner selected". The other items contain the names, in alphabetical order, of the runners who have already been recorded in the Runner table.

If a runner is selected from the dropdown, the time field is automatically calculated based on the distance field and the runner's speed.

At the same time, the total distance field in the selected runner's row in the Runners table is updated (in all cases it should contain the total length of the distance the runner has to run).

All of the fields except the runner field are read only and should not be modifiable by the user.

The calculator app must ensure that the data entered by the user is automatically saved locally, so when the browser is restarted or the page is reloaded, the previously entered data is displayed again.

Additional information

- Some media, icons and text have been provided for you in the media files. You are free to use these, but you can also create your own, as long as the website is still fit for its purpose. **You should not use any other media files (e.g. downloaded videos, images, icons, etc.)**.
- **You are allowed to use only the following Javascript frameworks: React, Angular, Vue.**
- You are not allowed to use any other JS or CSS frameworks or libs (e.g Ember.js, Bootstrap, Tailwind etc.)
- The use of any additional npm module is forbidden. You can only use npm modules which are installed by the default project generation process of the selected JS framework (create-react-app, ng new, vue create).
- However, you may use CSS prepocessors during development.
- Clean code and user interface accessibility are also important considerations.

- Please use semantic elements in HTML files wherever possible.
- Provide the necessary amount of comments in your HTML, CSS and JS files.
- Check accessibility using the Axe browser extension.

- The application must work with at least the latest versions of Google Chrome and Firefox.