

Web Technologies
WorldSkills 2024 National Competition
HUNGARY

Final
Module D – Frontend

Submitted by:

Skills IT

Contents

Introduction.....	3
Testing	3
General expectations on appearance.....	3
Description of tasks.....	4
Runner selection page	4
Tracking Page	5
<i>Welcome text.....</i>	<i>5</i>
<i>Current runner box</i>	<i>6</i>
<i>Next run section.....</i>	<i>6</i>
Appendix B – Available endpoints for Module D	9
My Team.....	9
Schedules.....	9
Handover	11
Utility endpoints	11

Introduction

In this module, you will create a mobile-optimized web application that runners will use to easily track events during the competition and to record handover times for their own runs.

The application communicates with our fully functional backend using REST API technology. You can use JavaScript framework (or vanilla JS) to build the application.

To perform this task, you have the backend already used in Module C (<http://backend2.ub2023-YY.hu/api/v1>). In addition to the previously used endpoints, additional endpoints required for the Runner app are provided by the backend. These are described in **Appendix B** at the end of this document.

Testing

To make testing easier, we provide a few endpoints to reset the database and simulate the different situations before, during and after the race:

- **Reset Database:** reset the database to the initial state with random stage assignments (/resetDb)
- **Reset Handover Times:** Remove all previously registered handover times (/resetHandover)
- **Race Simulation:** Generate handover times up to the specified stage. If this endpoint is called with e.g. stage=3 parameter, the backend generates realistic handover times for the first 3 stages. The handover time of the 3. stage will be the actual time. (This simulates the situation where the runner of stage 4 is currently running.). If you want to simulate that the team has finished running, you must call this endpoint with stage=54. (/setHandover)

You can find these endpoints in the backend2.postman_collection.json postman collection in the assets\postman folder.

You can also use the Stage Planner app (<http://stage-planner.ub2023-YY.hu>) to view or change the distribution of stages between runners. Changing the stage assignments on the Stage Planner app resets all handover times.

General expectations on appearance

- Since runners will be using the app on their mobile devices, you need to create a mobile-optimized version of the app. We will not test your work in desktop view. The primary testing environment is iPhone 12 Pro.
- During development, you should follow the design given in the mockups as much as you can. All content must be displayed in full screen without scrolling.

You can find all the mockups showing all the typical states of the app in the /assets/images/runner-app folder.

- As a help, you can find the style guide with colors of the application in the assets/style-guide/StyleGuide.jpg file.
- The name of the team is always displayed at the top of the window.

Description of tasks

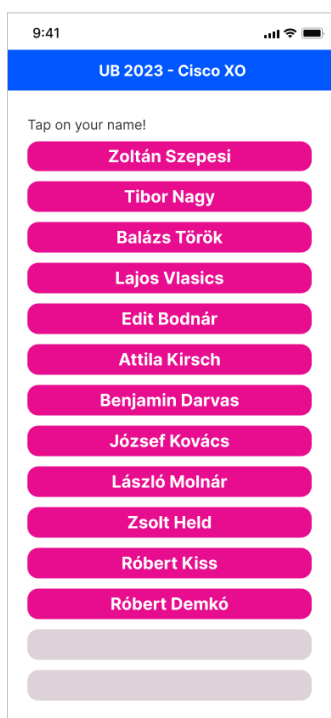
The application consists of two main pages:

- Runner selection page
- Tracking page

Runner selection page

If the team has been successfully identified with the access code entered in the URL (e.g. <http://runner-app.ub2023-vy.hu/?accesscode=123456789>), this page will be displayed first.

On this page you can see the names of the team members. There is a maximum of 14 team members, so a total of 14 buttons should be displayed. If the team consists of less than 14 team members, an inactive placeholder should be displayed in the remaining space as shown in the mockup below.



Clicking on any team member's name will display the Tracking Page.

Tracking Page

The tracking page can track the race from the perspective of the selected runner.

The information displayed on the Tracking Page changes according to the stage of the team and the selected runner's race.

The page consists of the following main parts:

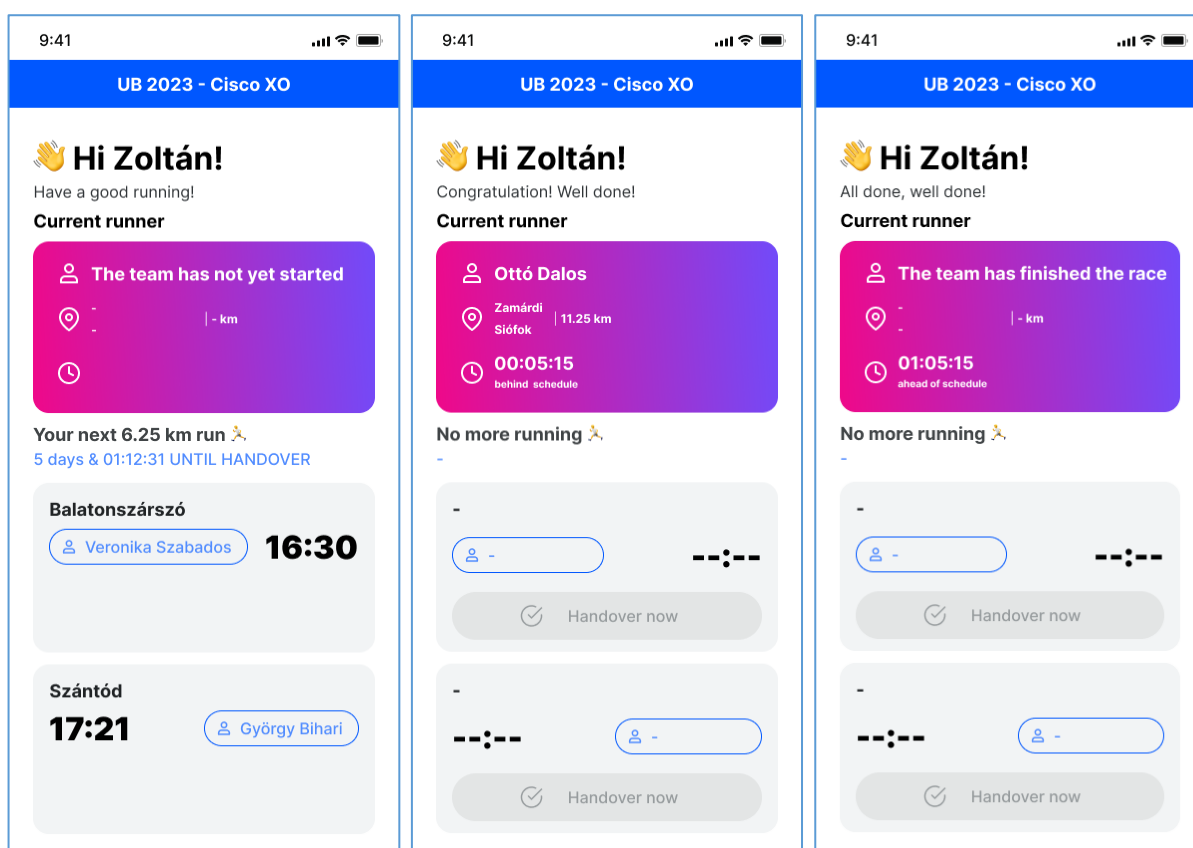
- **Welcome text**
- **Current runner box**
- **Next run part**

Welcome text

At the top of the screen a welcome text with the first name of the runner appears. E.g.: *“Hi Zoltan! Have a good running!”*

When the selected runner has finished their last stage, the second sentence will be changed to *“Congratulation! Well done!”*

When the last runner in the team crosses the finish line, the text will read: *“All done, well done!”*



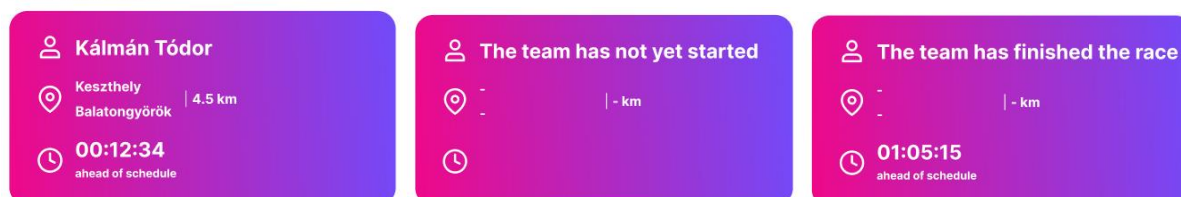
Current runner box

Once the team has started its running, the top third of the window shows the details of the runner currently running (runner name, stage starting and arriving location, stage length time difference compared to the original schedule). This data is supplied by the backend on the **/currentRunner** endpoint. The time difference compared to the original schedule is returned in seconds in the "scheduleDifference" field of the endpoint. A positive value means that the team is ahead of schedule, a negative value means that the team is behind schedule.

Display the data as shown in the designs below.

If the team has not yet started the race, *"The team has not yet started"* is displayed instead of the name. If the team has already finished its race, the text *"The team has finished the race"* is displayed.

Current runner



Next run section

You can get all the necessary details of the runner's next run using the **/nextRun** endpoint.

What data does **/nextRun** return?

The endpoint returns data in the following format:

```
{
  "stage": {
    "id": 13,
    "startingLocation": "Badacsonytördemic",
    "arrivalLocation": "Szigliget",
    "distance": 3.4,
    "name": ""
  },
  "previousRunner": {
    "id": 3,
    "firstName": "Tibor",
    "lastName": "Nagy",
  }
}
```

```









    "pace": "06:00",
    "teamId": 1
  },
  "nextRunner": {
    "id": 1,
    "firstName": "Zoltán",
    "lastName": "Sisák",
    "pace": "06:00",
    "teamId": 1
  },
  "canStart": false,
  "plannedStartTime": "2023-03-31T19:51:39.401055Z",
  "plannedFinishTime": "2023-03-31T20:12:03.404243Z"
}

```

As you can see, this includes the stage of the runner's next run, the **previousRunner** includes the details of the runner from whom he is taking over the baton, and the **nextRunner** shows the runner he is switching to at the end of his run.

canStart becomes true when the previous runner has already started his/her run.

The planned start time (**plannedStartTime**) and the planned finish time (**plannedFinishTime**) are recalculated by the backend when a handover occurs and the time of this handover is entered in the database.

<p>Your next 6.25 km run 🏃</p> <p>00:12:31 UNTIL HANDOVER</p> <hr/> <p>Balatonszárszó</p> <p> Veronika Szabados 16:18</p> <p> Handover now</p> <hr/> <p>Szántód</p> <p>17:09  György Bihari</p> <p> Handover now</p>	<p>Your next 7.00 km run 🏃</p> <p>4 days & 03:15:00 UNTIL HANDOVER</p> <hr/> <p>Balatonfüred</p> <p> 06:00</p> <p> Start Race now</p> <hr/> <p>Aszófő</p> <p>06:42  György Bihari</p> <p> Handover now</p>
---	---

This section shows the details of the next run of the selected runner.

- Length of the runner's next run (stage) in km. If the selected runner finished all of their running (stages) "No more running" text should be displayed.

- Time remaining until the next run:
 - If there is more than 1 day to go, the total number of days remaining should also be displayed.
 - This counter should be updated continuously without refreshing the browser window. (See the description of the handovers below.)
- Details of the starting point of the stage
 - name of the starting location
 - the planned handover time
 - name of the runner from whom the selected runner will take over the baton (previous runner)
 - the button to record the handover, which is active and clickable if the value of **canStart** is true and inactive if **canStart** is false. Button should not be displayed if there is more than one day to go until the next run of the selected runner.
- Details of the finishing point of the stage
 - name of the finishing location
 - the planned handover time
 - name of the runner who will take over the baton (next runner)
 - the button to record the handover, which is active only if the selected runner already started his run on that stage and inactive otherwise. Button should not be displayed if there is more than one day to go until the next run of the selected runner.

If the selected runner will run the first stage (and has not yet started their run), the “Start race” button at the top of the screen will be displayed. For other runners the button will be “Handover now”.

If the first runner clicks on the “Start race” button, the start time will be added to the team's **startingTime** field by a post request to **/handover/start** endpoint. From this moment on, the backend calculates the **plannedStartTime** and **plannedFinishTime** values relative to this time (e.g. **/nextRun** endpoint)

When a team member finishes a stage, they must pass the baton to the next runner (we call this action the “handover”). For recording this action, there are 2 endpoints you can use: **/handover/start** and **/handover/finish**. These endpoints always receive your stage id (the one returned by **/nextRun**) and the selected runner's id.

The **/handover/start** records the time of the event when the previous runner hands over the baton to you. The **/handover/finish** endpoint records the time at which you hand over the baton to the next runner. (This means, that both the current and the next runner can press the handover button, the result will be the same, but only one of them needs to click it.)

To keep the UI up to date, the **/nextRun and **/currentRunner** endpoints need to be polled every 10 seconds, and after every handover action.**

Appendix B – Available endpoints for Module D

The following endpoints are available under <http://backend2.ub2023.hu/api/v1> for you. You do not need to implement any of them. All endpoints (except the utility ones made for testing) need to be provided with an Authorization HTTP header containing the accessCode of the team. The initial database contains an accessCode of 123456789 for team #1.

My Team

- **GET /myteam**

Returns information about the team and its members.

- Required header: Authorization: Bearer <accessCode>
- Possible responses:
 - 200 OK

```
{
  "id": 1,
  "name": "Team 1",
  "contactEmail": "team1@example.com",
  "plannedStartingTime": "2023-04-25T09:20:00.000Z",
  "startingTime": "2023-04-25T09:21:05.123Z",
  "runners": [
    {
      "id": 10,
      "firstName": "Tódor",
      "lastName": "Kálmán",
      "pace": "05:31",
      "teamId": 1
    },
    ...
  ]
}
```

Schedules

- **GET /currentRunner**

Returns information about the runner currently running and the stage they are on. Also returns the time difference relative to the planned schedule.

- Required header: Authorization: Bearer <accessCode>
- Possible responses:
 - 200 OK

```
{
  "runner": {
    "id": 10,
    "firstName": "Tódor",
    "lastName": "Kálmán",
    "pace": "05:31",
  }
}
```

```

    "teamId": 1
  },
  "stage": {
    "id": 10,
    "distance": 3.5,
    "startingLocation": "Balatonakali",
    "arrivalLocation": "Fövényes",
    "name": "NN"
  },
  "scheduleDifference": -1362
}

```

- 200 OK (if all members of team finished, the race is completed for the team)

```

{
  "finished": true
}

```

- **GET /nextRun?runnerId=:runnerId**

Returns information about the upcoming stage (run) of the runner provided in the URL query parameter. If the team member is currently running, it returns the current stage (meaning that it always returns the first non-finished stage assigned to this runner).

- Required header: Authorization: Bearer <accessToken>
- URL Query parameter: runnerId
- Possible responses:
 - 200 OK

```

{
  "previousRunner": {
    "id": 10,
    "firstName": "Tódor",
    "lastName": "Kálmán",
    "pace": "05:31",
    "teamId": 1
  },
  "nextRunner": {
    "id": 12,
    "firstName": "Jakab",
    "lastName": "Gipsz",
    "pace": "05:31",
    "teamId": 1
  },
  "stage": {
    "id": 10,
    "distance": 3.5,
    "startingLocation": "Balatonakali",
    "arrivalLocation": "Fövényes",
    "name": "NN"
  },
  "canStart": true,
  "plannedStartTime": "2023-03-31T11:51:39Z",
  "plannedFinishTime": "2023-03-31T12:06:52Z"
}

```

- 200 OK (if the runner is finished, has no more assigned stages to run)

```

{
  "finished": true
}

```

Handover

- **POST** /handover/start

Finishes the previous stage and starts the current one. It also starts the race, if the first stage is provided as an input.

- Required header: Authorization: Bearer <accessCode>
- Request body:

```
{
  "stageId": 1,
  "runnerId": 2
}
```

- Possible responses:

- 200 OK

```
{
  "success": true,
  "message": "Handover successful"
}
```

- **POST** /handover/finish

Finishes the current stage and starts the next one.

- Required header: Authorization: Bearer <accessCode>
- Request body:

```
{
  "stageId": 1,
  "runnerId": 2
}
```

- Possible responses:

- 200 OK

```
{
  "success": true,
  "message": "Handover successful"
}
```

Utility endpoints

These endpoints are made to make testing your frontend application easier. You can use Postman to invoke these endpoints during development.

- **POST** /resetDb

Resets the database to the initial state, randomly assigning team #1's runners to the stages.

- **POST** /resetHandover

Removes all handover times and starting times.

- **POST** /setHandover

Generate handover times up to the specified stage for the specified team. If this endpoint is called with e.g. stage=3 parameter, the backend generates realistic handover times for the first 3 stages. The handover time of the 3rd stage will be the actual time. (This simulates the situation where the runner of stage 4 is currently running.). If you want to simulate that the team has finished running, you call this endpoint with stageId: 54 (the id of the last stage).

- Request body:

```
{  
  "teamId": 1,  
  "stageId": 3  
}
```