Test Project

Module D Lyon Mobile Web Service

Web Technologies

Independent Test Project Designer: Thomas Seng Hin Mak SCM
Independent Test Project Validator: Fong Hok Kin

WSC2024_TP17_actual_en

Contents

Introduction

In this project, we are going to create a mobile web that provide essential Lyon services data. The web queries the dynamic data via a mock API server. This API server is pre-built and provided to you.

The server is PHP-based. You may find it in the dl.worldskills.org/module_d_api_server.zip. You will see a single file named module_d_api.php after extracting the zip file.

Please put this mock API PHP file into wsXX.worldskills.org/module_d_api.php

Description of project and tasks

Mobile web layout

There is header bar, main content, and navigation bar.

The header bar is always stuck to the top of the screen.

The navigation bar should be always stuck to the bottom of the screen.

When the main content is longer than the viewport, the scrolling happens only to the main content. The header bar and navigation bar stays.

The header bar may display the current view title, and provide navigation to go back.

Navigation Bar

There are 4 buttons in the navigation bar:

1. Carparks
2. Events
3. Weather
4. Setting

The Data API

You may find three major parts in the API server.

They are:

- wsXX.worldskills.org/module_d_api.php/carparks.json
- wsXX.worldskills.org/module_d_api.php/events.json
- wsXX.worldskills.org/module_d_api.php/weather.json

Paging

The events API has pagination.

You may find the next page URL in the following structure:

```
{
  "pages": {
    "next": "xxxxx",
    "prev": "xxxxx"
  }
}
```

Carpark availability

The first button in the navigation bar, is the carpark availability.

The mobile web page should display a list of carpark records. In each record, there are availability count.

Sorting

There is a toggle for switching between sorting by alphabet, or sorting by distance between the current location to each carpark. The toggle setting in inside setting view—the fourth button in navigation bar.

Geolocation and Simulating geolocation

We need to simulate a location.

There are two methods: Manually set the geolocation, or using Chrome's geolocation dev tool.

The geolocation can be manually set by the URL query: latitude and longitude. For example, user can append *?latitude=45.755051&longitude=4.846358* to the URL to simulate a geolocation.

The JavaScript code to calculate geolocation distance is provided.

The mobile web should be able to request for the current geolocation as the default method. And due to the security policy and the limitation of the competition environment, if the JavaScript code of getting geolocation is blocked by web browser from running due to competition environment limitation, we may assess the source code for the geolocation implementation.

Focusing on a carpark

User can click on a carpark in the list to focus on one carpark.

When focusing on a carpark. The view only displays the carpark name, distance, and the availability number.

User can go back to the carpark list from this view.

A carpark can be pinned to top

Carpark can be pinned and unpinned to the top of the list, regardless of sorting method.

The pinned status should be saved to local storage and be restored on page refresh.

Lyon Events

The second navigation button links to the Event List.

The list displays some events initially, according to the default API returning data. The events are listed from top to bottom. For each event record, there is image, event title, event date.

Querying the events data

The event list can be filtered by beginning and ending date.

There is an input to select a beginning date, and an input to select an ending date.

When the beginning date, and/or ending date, are selected. The event list refreshes to list the updated query result.

/module_d_api.php/events.json?beginning_date=YYYY-MM-DD&ending_date=YYYY-MM-DD

Infinite scrolling

When the user scrolls to the bottom, the web page should load further event data and display further events to the users. This is also known as infinite scrolling.

Please fine-tune the timing and smoothness of infinite scrolling. The loading of further content should be not too late, and not loading too early. In other words, the infinite scroll should not feel laggy, and should not eagerly load too many data. Please ensure the data loading is correct. That means there should not have multiple instances of the same record displayed, and not missing any records to display.

Weather

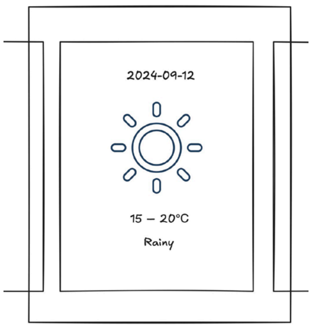
The third navigation button is for weather.

It shows the up-coming week weather. There are 7 days of weather.

Horizontal Scrolling

The week weather is aligned horizontally. When scrolling horizontally, the weather snap to the day.

Please refer to the weather mockup in the media file.



SVG

There are SVG icons for each day.

The SVG icons are based on the weather status of the day. The responded JSON data have defined which icon to use.

When mouse hovers on the SVG icon, the SVG has an animated stroke effect.

The SVG has stroke color #1c3e60, with stroke width 1 and none fill.

It has stroke dash-array style from 50 to 200 for 2 seconds, and stroke dash-offset style from 200 to 0.

Please refer to the svg-icon-animation.mp4 in the media files.

Setting

The fourth button in the navigation bar is the setting view.

Dark theme

There is a configure to switch the theme between dark theme, light theme, and an option to follow the system's light/dark theme setting.

Carpark Sorting method

There is a toggle for switching between sorting by alphabet, or sorting by distance between the current location to each carpark.

Instructions to the Competitor

Mobile Web App configuration

Please configure the manifest.json and essential meta tags for both Android and iOS mobile web system.

The viewport should also be configured for mobile web friendly viewing.

Accessibility

Please ensure having good accessibility. We will assess the Chrome Lighthouse accessibility score.

Please also ensure creating easy-to-maintain JavaScript code.

Other

This project will be assessed by using Google Chrome web browser.

You may provide a README file for executing guide if necessary.

Note if you are using NodeJS, please be aware of the node_modules files for Windows (on workstation) and Linux (on server) is different. Using a wrong node_modules folder may result in unexpected error.

Marking Summary

	Sub-Criteria	**Marks**
1	Mobile Web General	2.5
2	Carpark	3.0
3	Events	3.25
4	Weather	1.75
5	Settings & General	2.25