Middlesex
University
Dubai

Chinnu Mary George

Computer Engineering and Informatics

Coursework 1

Final Submission - Week 8 Friday

Web-Based Mobile App Development

CST3145

Module Leader(Hendon): Kai Xu

Module Coordinator (Dubai): Chinnu Mary George

8.2.1. Assessment 1(Web App with Vue.js)

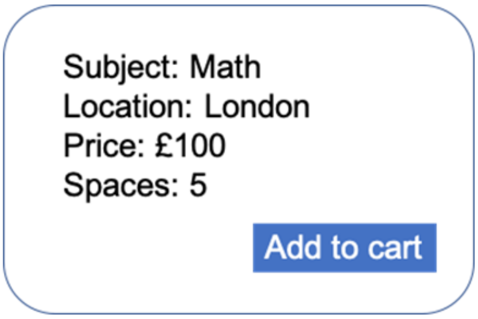
Web App with Vue.js	
Module code	***CST3145***
Module title	Web-Based Mobile App Development.
Submission deadline date, time	CW 1 (to be marked at LAB) Part1: Week6(GW) Part2: Week 8
Feedback type &date	Students will be given feedback and marked by the lab tutor.
Task	For this coursework, you need to create the front-end of a fictitious web app that allows students and their parents to look for after school classes and activities. The second coursework will add the backend and make the app work offline, and the third coursework will turn it into a mobile app.
Assignment type	The web app must be demonstrated
Requirements	The coursework does not require any backend storage such as a (MongoDB) database. All the data must be stored locally using local storage. You can use external CSS library such as Bootstrap. Make sure the library file or online link is included in the submission. Any JavaScript library is not allowed if it duplicates or replaces features provided Vue.js. Check with the tutor if not sure. **A submission will receive zero mark if it fails any of the requirements below:** • It must be implemented using *vue.js* framework; • The code must be stored in a *GitHub* repository with at least 10 commits; • The app must be available online using *GitHubPages*; • The web app must be demonstrated. **Coursework code will not receive any mark, even it works fine if**: It cannot be explained satisfactorily during the in-lab demonstration, i.e., student can- not explain what the code does. It is not implemented with Vue.js when it is possible to do so. This includes using plain JavaScript or other libraries.
Assessed learning outcome (s)	*1,2*
Assessmentweighting %	*35%*

Key reading and learning resources	This module has a variety of learning resources available for you to use to support your learning. These include recorded lecture, lecture slides, feedback, and key reading mate-rials. These can be accessed online via the module page. Please visit the module page regularly to make use of these. Your online reading list can be accessed from the My Study area of UniHub (http:// readinglists.mdx.ac.uk/lists/78D0F586-A45D-60DB-32A7-5D5EC274302B). This highlights recommended reading for this module. The course website has many links to other online resources.

Assessmentmarking criteria rubric (CW1)

Group in-lab tasks (10%): two tasks for each group, 5% each.

Group in-lab tasks (10%):



Subject: Math
Location: London
Price: £100
Spaces: 5

Figure 1: Coursework 1 - lab work

• Display the information of one lesson (3%):

– The information must include at least: subject, location, price, and number of spaces;

– All the lesson information must be stored in a Vue object;

- The information must include at least: subject, location, price, and number of spaces;
- All the lesson information must be stored in a Vue

object; The lesson must have an ‘Add to Cart’ button (4%):

- The ’Number of Space’ is reduced by one after clicking the ‘Add to Cart’ button if it was greater than zero;
- No change happens after clicking the ‘Add to Cart’ button if the ‘Number of Space’ equals zero.
- The interaction must be implemented using v−on.

The code is stored GitHub repository and can be accessed online using GitHub Pages (3%):

- The code must be stored in a GitHub repository with at least 5 commits;
- The Vue.js app must be accessible through GitHub

Pages.

- Individual task (25%):

Display lessons (4%)

- There should be at least 10 lessons and each lesson has 5 spaces;
- Each lesson should have at least:
  - Subject;
  - Location;
  - Price;
  - Space (how many spaces are left);
  - A image or icon.
- The list of lessons must be stored as an array of JSON objects, one object for each lesson, in a separate JavaScript file, such as lessons. js;

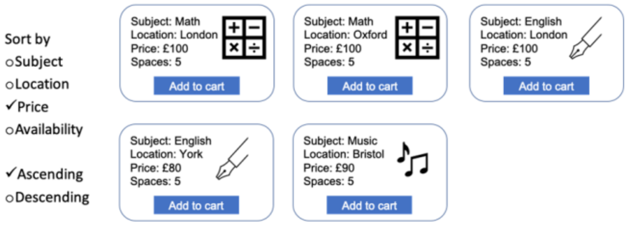
- v−for has to be used for the display of the lesson list.
  - Sort (4%)
  - User can choose to sort the lessons by one of the following attributes: subject, location, price, or availability;
  - There must be an option to sort in ascending or descending order, regardless of the attribute selected;

Add to cart (4%)

- Each lesson must have a ’Add to Cart’ button;
- The button is only visible when there is still space available;
- Clicking the button once will add one space to the shopping cart, reducing the remaining space by one;

Shopping cart (4%)

- The shopping cartbutton should only be visible after at least one lesson is added to cart;


Sort by
○Subject
○Location
✓Price
○Availability
✓Ascending
○Descending

Subject: Math
Location: London
Price: £100
Spaces: 5

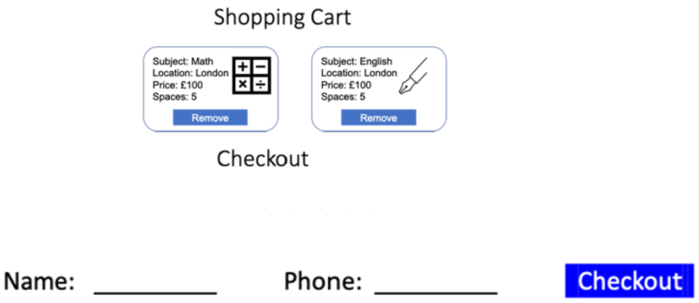
Subject: Math
Location: Oxford
Price: £100
Spaces: 5

Subject: English
Location: London
Price: £100
Spaces: 5

Subject: English
Location: York
Price: £80
Spaces: 5

Subject: Music
Location: Bristol
Price: £90
Spaces: 5

Figure 2: Coursework1 - Lesson List and Sort


Subject: Math
Location: London
Price: £100
Spaces: 5

Subject: English
Location: London
Price: £100
Spaces: 5

Name: ________ Phone: ________

Figure 3: Coursework1 - Shopping Cart and Checkout

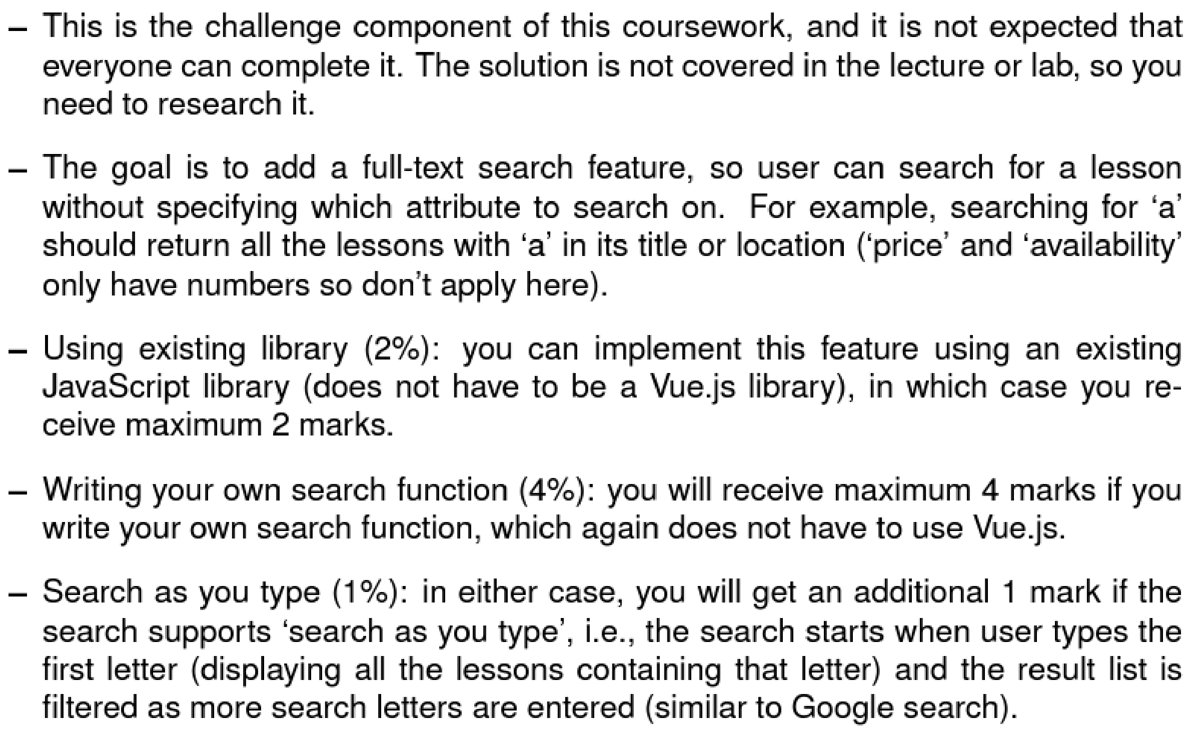
- Clicking the shopping cart button should go to a new page, with a button to go back to the lesson page;
- The shopping cart should show all the lessons added;
- User should be able to remove lesson from shopping

cart. Checkout (4%)

- The checkout is part of the shopping cart page;
- A user must provide the following information before he/she can check out:
  - * Name; *
  - * Phone;
- The ’Name’ must be letters only and the ’Phone’ must be numbers only;
- The ’checkout’ button is only visible after both valid ’name’ and ’phone’ are pro- vided;
- Clicking the ’checkout’ button should display a message confirming the order has been submitted.

Extra Challenge

Search (5%)



- This is the challenge component of this coursework, and it is not expected that everyone can complete it. The solution is not covered in the lecture or lab, so you need to research it.
- The goal is to add a full-text search feature, so user can search for a lesson without specifying which attribute to search on. For example, searching for ‘a’ should return all the lessons with ‘a’ in its title or location (‘price’ and ‘availability’ only have numbers so don’t apply here).
- Using existing library (2%): you can implement this feature using an existing JavaScript library (does not have to be a Vue.js library), in which case you receive maximum 2 marks.
- Writing your own search function (4%): you will receive maximum 4 marks if you write your own search function, which again does not have to use Vue.js.
- Search as you type (1%): in either case, you will get an additional 1 mark if the search supports ‘search as you type’, i.e., the search starts when user types the first letter (displaying all the lessons containing that letter) and the result list is filtered as more search letters are entered (similar to Google search).

Extension and Late Penalty

All extension must be applied through the Extenuating circumstances service (see Section 7.6 for more details). Please do not contact the module leader for extension.**The late penalty is 5% for each day after the deadline.** It is the 5% of your final mark. For example, if you receive 30 for your coursework

and are two days late, you coursework mark will be 30 (1 0:5 2) = 27.