COURSE PROJECT:
A SOCIAL MAP OF EVENTS

RELEASED: **10 NOV 2022 (THU)**
DUE: **17 DEC 2022 (SAT)**

SYNOPSIS

In a group of 5–6 students, you are going to set up a webapp to check cultural events or traffic speed for locations. Users will be able to log in and choose a subset of favourite locations. Your project app will retrieve location details from open datasets.

DATA SOURCE

You have to access some open API for actual data. You can pick among these two topics, based on your interest, ability, and preference:

1. **LCSD Cultural Programmes**
   https://data.gov.hk/en-data/dataset/hk-lcsd-event-event-cultural
2. **Traffic Data of Strategic / Major Roads**
   https://data.gov.hk/en-data/dataset/hk-td-sm_4-traffic-data-strategic-major-roads

If you identify a comparable dataset of location information from another source, you may propose to use it by emailing to chuckjee@cse.cuhk.edu.hk.

ACCESS MODES

Your app will provide two modes of access:

Users – Only authenticated users have access to the app contents. A user is recognized using a username (unique string of 4–20 characters) and password (string of 4–20 characters, *hashed*) pair. The user will be able to see and search for location details, maintain a group of favourite locations, and leave comments on locations.

Admins – Admins will be able to perform arbitrary CRUD actions to the location data and the user data on your database.

THE DATA

There are some differences in pre-processing of data from the two given sources for our local storage and display:

1. ***Dataset 1: Cultural Programmes***
   - Mainly consider the "Programme information" dataset
   - Data is available as XML
   - Pick only 10 venues to be shown in your app (*where each should host at least 3 events*)
   - Handle the following data: *title, venue, date/time, description, presenter, price*
2. ***Dataset 2: Traffic Data***
   - Mainly consider the "Traffic Speeds of Road Network Segments (Processed Data)"
   - Data is available as XML
   - Pick only 15 roads to be shown in your app (roads with a name, but not "5", *also not only a single segment*)
   - Handle the following data: *speed — show a range (min–max) if multiple segments found*

You need to *get the real-time information from API to database only once,* when the user *logins and loads your page.* Show the last updated time clearly. *No auto-update is required after that.*

For the project purpose, you are only required to prepare a number of locations in your app to be seen during project demo. It does not matter if you store or show more. You need to design the data schemas and models for storing (caching) items. For the locations, you are required to maintain at least:

- Venue name / road name
- Latitude and longitude

Only English data is required for the project app. For the schema and models for users and other data, you may design freely to suit your needs.

You may need to consult data dictionaries and related data for location details, and can feel free to use extra APIs for your app. ***Never use anything more than FREE TIER***!

APPLICATION REQUIREMENTS

User actions:

1. List all locations in a table as links to single locations, and allow sorting of the table with ***i) number of events at venue**, or* ***ii) min and max traffic speed of road***
2. Show all locations in a map, with links to each single location
   [Suggested APIs: Google Maps, MapBox]
3. Search for locations which contain *keywords in the name* which will result in a table of location results

4. A separate view for one single location, containing:
    a. a map showing the location
    b. the location details (events or traffic speed)
    c. user comments, where users can add new comments seen by all other users
5. Add location into a list of user's favourite locations, and see the list in another view
6. See the username in the top left/right of screen, and be able to log out

Admin actions:

1. ~~Request updated data of events or road names, i.e., reload from the online dataset, without affecting data which does not come from API (e.g., user comments within your app)~~ [Note: This requirement is cancelled]
2. CRUD stored event details or road names in the local database
    - *We will not test other features (e.g., map, comments) if deleting an existing location*
3. CRUD user data (username and password only) in the local database
    - *We will not test other features (e.g., comments) if deleting an existing user*
4. Log out as admin

Non-user actions:

1. Log in as user with username and password
2. Log in as admin using username and password both as `admin`

You may introduce pagination if you see fit, but it is not a requirement.

EXTRA REQUIREMENTS FOR GROUPS WITH ESTR2106 STUDENTS

Groups can be formed freely among CSCI2720 and ESTR2106 students.

- When 2 or more ESTR2106 students are involved, include all the following features
- When only 1 ESTR2106 student is involved, include at least two of the followings

1. *Charting Data*

Using any JS charting library (e.g. Chart.js), include one chart:

- Data set 1: Event price distribution of a venue
- Data set 2: Traffic speed history (average of each hour) in the past 24 hours

2. *GraphQL API*

With an endpoint `/graphql`, these queries should be supported with real-time data:

- List all location details (including DB data, API data, and user comments)

- Single location with details (including DB data, API data, and user comments)

A clear guide to use the API should be provided somewhere in the project site.

3. ***Multiple colour schemes***

At least 2 visually different colour schemes should be supported in the project site. It can be chosen by the user, and be saved as a user preference, and reappear every time when logging in.

4. ***Server log in Node***

With the Node server backend, produce a log text file for all the frontend requests. The log file should contain each request in a separate lines, with these details:

- User IP
- User browser info
- Date and time (timezone = GMT+8/HKT)
- Request method and URL

SYSTEM REQUIREMENTS

You are strongly recommended to host the project on the AWS EC2 virtual machine in your assigned free account, owned by *one of the members*. Details are provided in *Lab 9*. The following facilities are provided:

- *Node v18.12.1 + npm 8.19.2*
- *MongoDB server 5.0*

The backend used should be Node. The frontend platform is not restricted for the project. Yet, your project app must be a ***Single Page Application***, without refreshing the page for any internal links. However, visits to all different views should be reserved in the browser history, with a proper URL.

You will decide the complexity and aesthetics of your work. Make sure it is clear and useable by average users (e.g., your TAs) without much guesswork. You can make decision on anything not specified in this document, and extend beyond the basic requirements.

You may freely decide the choice of technologies and frameworks to be used in this project. However, make sure that the site should be running on *production build* instead of development mode.

The grading will be done on a desktop/laptop computer using Google Chrome (*almost-newest versions*), so your app should at least serve HTML and relevant styling and scripting codes.

ASSESSMENT AND SUBMISSION

All technical features would be graded during the demo. Your project will be graded by:

- Technical requirements – *fulfilment* and *complexity* (50%)
- Usability – *look and feel* (20%)
    - *This includes whether a smooth SPA experience is provided, with responsiveness in layout and clarity of text/colour presentation*
- Project demo (10%)
- Project outline/report (20%)

1. PROJECT OUTLINE [**NOV 21**, 23:59]

Submit a one-page document discussing the followings:

- Group members, data source, data schema plan, APIs to be used, reasons for your chosen platform and technologies comparing to others, and anything you consider relevant

2. PROJECT DEMO [**DEC 17** AFTERNOON] AND SUBMISSION [**DEC 17**, 23:59]

Details on the project demo will be announced at the middle of December. During the demo, a public IP needs to be used to access the project site at port 80 by the demo grader.

Include full names and student IDs of all members in ***all code files*** using comments. Zip all your files and submit it on the course site on Blackboard.

You do not need to submit the* `node_modules` *folder,
***but please keep the files* `package.json` *and* `package-lock.json`.**

Submit also a **`readme.txt`** to state the project server start commands, as well as your site URL. Inside the file, indicate clearly whether or not you have read this article carefully: *http://www.cuhk.edu.hk/policy/academichonesty* and include the required declaration.

3. PROJECT REPORT [**DEC 23**, 23:59]

You need to submit a document to describe your project, ***with reference to the CSCI2720 course materials***, and other online references. Here are suggested components for your report:

- **Abstract**
  - A summary of your work in no more than 100 words, with one screenshot of a representative screen of your site
- **Methodologies**
  - Discussion on the programming language(s) and important algorithms you have used (and you may reference the course materials if needed)
  - Design of data schemas and models of your database
  - Files submitted to Blackboard
  - Description of all libraries/frameworks used, other than HTML/CSS/JS
  - A comparison table of at least two advantages and two disadvantages (specific to your project app) of your chosen platform and technologies comparing to others, e.g. *"Why MySQL?"*
- **References**
  - Citation of all materials which are not originally written by you, including teaching materials in and out of our course
  - You must use the IEEE style properly
    (*Ref: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/conferences/style_references_manual.pdf*)
- **Appendix**
  - Anything you consider supplementary, e.g. workload distribution

You may feel free to include more figures for this report.

Please use 4–6 pages for the report*,* with 11pt 1.5 line spacing. Penalties will be applied for anything out of the allowable range. The project report also needs to be submitted to VeriGuide (*https://academic2.veriguide.org/cuhk*) *by one of the group members*.