CSCI 2720/ESTR2106 – Building Web Applications

Course Project

Released: 5h November 2025

Submission deadline: 23:59:00, 18th December 2025

Synopsis:

Your group is going to set up a web application to check information on some locations. Users and Admin will be able to log in and perform certain actions. Your project app will retrieve location details from an open dataset. 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.

The extra tasks for ESTR2106 students are highlighted in red.

Data source:

A suggested dataset for this project:

Cultural Programmes https://data.gov.hk/en-data/dataset/hk-lcsd-event-event-cultural

You can choose the database that interests you. The only restriction is that it must be publicly available real-world location data. If you identify a comparable dataset from another source, you may ask Dr Colin Tsang (colintsang@cuhk.edu.hk) for permission.

System requirements:

Your project will be graded with the following:

- Google Chrome (almost latest version)
- Node v24.9.0 + npm 11.6.2
- MongoDB server 8.0.13

The backend use should be Node. The fronted platform is not restricted for this project. You are allowed to utilize additional methods beyond what has been covered in our lectures and labs. However, please note that you cannot expect our TA to use any external software or tools to run your files. During the grading process, we will only use the above environment.

The data:

You need to perform pre-processing to the dataset for the local storage and display:

Dataset: 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.

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.

You need to design the data schemas and models storing (caching) items. For the locations, you are required to maintain at least:

- Location 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 location details, and you can feel free to use extra APIs for your app. ***Never use anything more than Free Tier.***

Access modes:

Your app will provide two modes of access:

1. ***Users*** – only authenticated users have access to the app's contents. A user is recognized using a username and password pair. The user will be able to perform the "user actions", which are specified on the next page.
2. ***Admins*** – admins will be able to perform arbitrary CRUD actions to the location data and the user data on your database.

Basic application requirements:

User actions:

1. List all locations in a table as links to single locations and allow sorting the table with location names, distances, and the number of events at venue.
2. Show all locations in a map, with links to each single location (suggested APIs: Google Maps, OpenStreetMap or MapBox).
3. Filter locations by keywords, areas, and distance (e.g., within x km), with dynamic updates to the location list and map without page refresh.
4. A separate view for one single location, containing:
    a. A map showing the location.
    b. The location details.
    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-right of screen and be able to log out.

Admin actions:

1. CRUD stored event details in the local database.
    a. We will not test other features (e.g., map, comments) if deleting an existing location.
2. CRUD user data (username and password only) in the local database.
    a. We will not test other features (e.g., comments) if deleting an existing user.
3. 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.

Extra features:

You should add at least one extra feature on top of the basic requirements. You are free to design any extra features. Below are some suggestions for your reference:

1. Adding a dark/light theme switch button and, optionally, match the website theme with the user's system theme.
2. Including a like button, similar to Facebook, for users to interact with events.
3. Using an HTML form to gather user information and provide event recommendations accordingly.
4. Implementing an event booking system for events.
5. Security.
6. Adaptive layout for different screen sizes and/or mobile devices.
7. Internationalisation (i18n) and localisation (l10n).

It's not necessary to follow my suggestions above. Feel free to design any appropriate feature for your website. **For ESTR2106 groups, you must include at least three additional features.**

Project report:

You need to submit a report to describe your project. The main purpose of this report is to help your grader (i.e., our TAs) understand your web page better. The report should be simple and easy to read. Anything you consider important should be included in this report. Here are suggested components for your report:

1. ***Abstract:***
   - A summary of your work in no more than 100 words, with one screenshot of a representative screen of your site.
2. ***Methodologies:***
   - List the files submitted to Blackboard with short descriptions.
   - Discussion on the pre-processing of the dataset.
   - Discussion on all required actions of your app.
   - Discussion on the extra features designed by your group.
   - Discussion on the important algorithms/functions/components you have used.
   - Design of data schemas and models of your database. Figures are suggested.
   - **For ESTR2106 groups, please include a list to indicate which ESTR2106's contents are applied in this project. For example, TypeScript, customized Bootstrap, functional states, lazy loading, etc.**
3. ***Contact Person:***
   - Please select one student as the contact person. We will contact this student if we have trouble opening your file during the grading period.
4. ***References:***
   - Citation of all materials which are not originally written by you.
   - You must use the IEE style properly: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/conferences/style_references_manual.pdf
5. ***Appendix***
   - A workload distribution. We allow more than one person to do a job together.
   - Anything you consider supplementary.

You are recommended to include more figures for this report. Please ***use 5-10 pages for the report, with 11pt 1.5 line spacing***. Penalties will be applied for anything out of the allowable range.

Assessment:

Your project will be graded by:

1. Technical requirements (60%)
   Fulfilment of basic requirements.
2. Usability – look and feel (30%)
   Your extra features will be considered here. This includes whether a smooth SPA experience is provided, with responsiveness in layout and clarity of text/colour presentation. Like Problem 0 of Assignment One, a well-designed website with simple extra features typically falls in the 15-25 pts range. We will not give out 30 pts easily. Achieving full marks requires an exceptionally designed website with impressive extra features.
3. Project report (10%)
   Provide good discussion and demonstration of your project. Highlight your extra features and anything you consider important.

The project is designed to resemble an examination. To maintain fairness, we will not offer additional assistance or guidance. We will only clarify the specifications if they are unclear.

You will decide the complexity and aesthetics of your work. Make sure it is clear and useable by any users (e.g., your TAs who will grade this project) without much guesswork. You can decide on anything not specified in this document. You may freely decide the choice of technologies and frameworks to be used in this project, as long as it is runnable on our grading environment specified on the first page of this document.

Submission:

Include full names and student IDs of all members in all code files using comments. Zip all your files into a single zip file:

GROUP_[your group number].zip

Select one student as a representative to submit it to 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*** (inside the zip file) to state the project server start commands, as well as your site URL. Inside the file, indicate clearly whether you have read this article carefully: https://www.cuhk.edu.hk/policy/academichonesty/ and include the required declaration.