

CSCI 2720 – Building Web Applications

Course Project Report: Group 40

Lau Ho Man 1155157519
Ho Ka Chun 1155176593
Lam Cheuk Wing 1155158991
Lam Hoi Chun 1155192755
Lee Chi To 1155143279

1. Abstract:

● A summary of your work in no more than 100 words, with one screenshot of a representative screen of your site.

We have built a system with a login function and listing information for Cultural Programmes, including their locations and events.

Also, Clients are classified into user and admin.

For users, they can check the information of the programmes in the user page. In addition, they can search the location, click the link to see information about the location and look at the venue location on Google Maps.

For admin, they can CRUD user account (Admin account is hard-coded) and CRUD events on the admin page.

Locations Table

Location	Number of Events
Shau Kei Chung Centre (Lecture Hall)	28
Shau Tin Tuen Hall (Conference Room)	23
Shau Tin Tuen Hall (Lecture Room 1)	17
Kai Shas Theatre (Theatre)	14
Hong Kong City Hall (Theatre)	9
Hong Kong City Hall (Exhibition Gallery)	9
Hong Kong Cultural Centre (Foyer Exhibition Areas)	8
Hong Kong City Hall (Exhibition Hall)	6
New Chi Wan Civic Centre (Cultural Activities Hall)	4
Hong Kong Cultural Centre (Exhibition Gallery)	4



Location Search

Hong Kong

Results

Location	Latitude	Longitude
Hong Kong City Hall (Theatre)	22.282279	114.161545
Hong Kong City Hall (Exhibition Gallery)	22.282279	114.161545
Hong Kong City Hall (Exhibition Hall)	22.282279	114.161545
Hong Kong Cultural Centre (Foyer Exhibition Areas)	22.29386	114.17053
Hong Kong Cultural Centre (Exhibition Gallery)	22.29386	114.17053

2. Methodologies:

- List the files submitted to Blackboard with short descriptions.
 - /dist: The folder to serve all files needed while running the web application, including all-in-one javascript and html file
 - /Server/server.js: Contain node.js code, and all backend functionality, including response to a request and database access.
 - /src/index.html: The base html file for rendering the web application.
 - /src/index.js: The javascript file that mainly used React.js to render the appearance of the web application.
 - /.babelrc: The babel file to define the configuration for translating react.js to javascript

- /webpack.config: The webpack library configuration for setting up the web application development and build.
- /package & package-lock: Contain all node modules used and their configuration.

● Discussion on the pre-processing of the dataset.

All pre-processing of the dataset is performed inside the function: `initEventInfo()`

1. Fetch all locations from Venues of programmes
 - a. Use `window.DOMParser().parseFromString()` to parse the XML response as HTML collection
 - b. construct location object with properties: {location ID, location name, location latitude, location longitude}
 - c. IMPORTANT: The price of the recommended dataset is varying: missing / single price / multiple prices / string: free / string: free with tickets / string: free before a point of time and pay after a point of time. To simplify the work, we decided to arrange those missing price as \$0 and other case as a random price inside range [0, 200]. Those the price may not be the same as the price shown in the dataset!
2. Fetch all events from Programme information
 - a. Use `window.DOMParser().parseFromString()` to parse the XML response as HTML collection
 - b. construct event object with properties: {event ID, event title, event venue, event date, event description, event presenter, event price}

(PLEASE note that not all events from the dataset contain all properties above, the absence of properties is represented by "None")

(ALSO, the date is string type instead of date type because the data is in string sometimes. The price is not number type due to the same reason)
3. Pick 10 venues from all locations that contain at least 3 events for each venue.
4. Filter all events, only left events from above 10 venues
5. Store processed events and locations in mongoDB and cache them in the state of `<UserPage/>`

● Discussion on all actions of your app, including the new features designed by you. E.g., explain if you have applied any restrictions on username and password.

For the admin page, there are two components to control events and user details. When the administrators log into the page, the page requests the server to obtain details of the events and users for the components so that the page can display all the details.

The lower component consists of several rows to let the admin see the list of events that have already been filled in the forms. In order to modify it. Administrators can modify the data inside one form to modify one event. Then they click the "update" button to send the request to the server. The server will according to the id sent by the client to look for the event record and modify it according to the form. Finally, the client refreshes the component after it receives the response. Administrators can also click the "delete" button near the row to delete the entire event. The client would send a request to the server with event id as the data. The server would look for the event according to the event id and delete the record in the database. Finally the client refreshes the component after it receives the response.

In order to create a new event record. Administrators can type the new data into the “new” form. Then they click the “create” button to send the request to the server. The server would base on the name of the location sent by the client, look for the location record to check if the location existed or not. If it exists, the server would create a new record based on the form and save it to the database. Finally the client refreshes the component after it receives the response.

The upper component is similar to the lower part but it displays user information instead of event information. Administrators would modify the details inside the form to update the user or create a new user. They can also click the “delete” button to delete the record.

When the server cannot find the required record when deleting the event, updating, or creating the event, it will respond to the client and the client will send an alert to the administrators so that they would be informed where the problem is.

For the user page, there are 8 functions that users can use or see.

To begin with, users can see their username at the top-right of the screen. Next to the username, there is a log-out button. After clicking the button, the page will log the user out to the login page. Furthermore, in order to provide more choices on the page’s look and mission for users’ eye strain reduction, we have set a button to switch dark mode and light mode. Light mode leads to a black color for text and white for the background, while vice versa for the dark mode.

Users can check all the event locations in a table showing such information. Users can search the event location that they want by typing keywords of the location name into the input box. The results for the keyword will be shown. A Google Map is provided for users to check the exact location of the event, which are pin points on the map. A table showing the number of events in different locations is provided. The table sorts the number of events.

Users can click on the locations in the location table which leads to a separate view that shows a map with the location, the location details(name, latitude, and longitude), a user comments section where users can comment and view others' comments stored in the database and a button for adding that location to a list of user’s favourite locations which should be shown in the user page.

For non-user functions, clients who enter the system should see a simple login form. The username and password inputted will be sent to the backend for validation by searching in the database. Clients can only log in if their username and password exist in the database. The root user (admin) is hard-coded in the server.js and will be initialized if not exist in the database. Last but not least, we also used a User schema which consists of a username and password. Both of them are string type and required, but only the username is unique.

● Discussion on the programming languages and important algorithms you have used.

We used react, html, javascript, node, css in the web application.

First, we use both html and react to construct the frontend for the system. We use a basic index.html for rendering react component in frontend. Besides, we use react component to construct tables, which have great reusability, passing props variable also facilitate the information communication among components, and caching state variable also avoid great network traffic between frontend and backend.

Second, we use css to decorate our front end. It allows better UI.

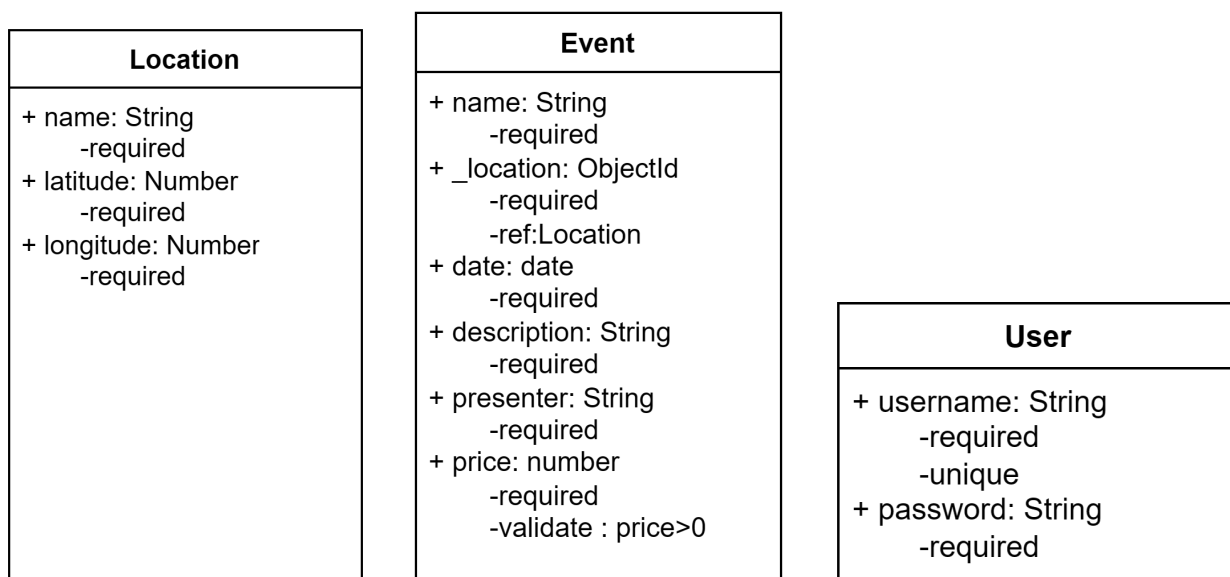
Third, we use javascript to handle inputs and information. For the login function, we submit it as a form to the backend. Also, it is used with fetch API to get information from the backend, including get events, locations, etc.

Fourth, we use node to build the backend for the system. We also used different node modules to facilitate the development of the backend, including express, mongoose, etc. For most of the code in server.js, we use node to handle requests from different routes, to return information from the database.

In addition, we used a Universally Unique Identifier algorithm in our system. It is used to set a unique ID for each TableRow.

● Design of data schemas and models of your database. Figures are suggested.

There are 3 data schemas in the database. They are “Location”, “Event” and “User”. Administrators can modify “Event” and “User” on the Admin page. The “_location” field is to link up the location and the event to record the location of the event.



● Description of all libraries/frameworks used, other than HTML/CSS/JS. (natural language is better, but some core codes are acceptable)

Libraries/frameworks used:

- Google Map API
 - used for generating maps and displaying the locations on the map
- React
 - to create Single Page Application and develop the page with different modules
- Mongoose
 - to perform CRUD action on MongoDB in a simplified way

- Node
 - to setup local server as a middleware to connect frontend and database, process information and control traffic.
- Webpack
 - to take all html and javascript files and bundle them into an optimized output file, known as bundles.
- Babel
 - serve as a JavaScript compiler to convert javascript into backward-compatible versions that can run on older browsers or environments.
- cors
 - avoid cors error

● A comparison table of at least two advantages and two disadvantages (specific to your project app) of your chosen platform and technologies compared to others.

Advantage / Disadvantage	Our chosen platform and technologies	Others
Advantage 1:	React: its component-based architecture allows us to break down the user interface into reusable components, which can be composed together to build complex UIs. This modularity makes it easier to maintain and update the codebase. For example, if a specific component needs to be modified or fixed, developers can focus on that component without affecting the rest of the application.	Angular: its comprehensive framework provides a complete solution for building large-scale applications. Angular includes features such as a powerful templating system, dependency injection, routing, and state management out of the box. This integrated approach can lead to increased productivity and reduced development time, especially for complex projects.
Advantage 2:	Node.js: its event-driven, non-blocking I/O model. This makes it highly efficient and well-suited for handling concurrent requests and real-time applications. Node.js uses JavaScript on both the frontend and backend, allowing for easier code sharing and improving developer productivity. Additionally, Node.js has a vast ecosystem of libraries and packages available through npm (Node Package Manager), which helps developers leverage existing solutions and accelerate development.	Django: it has great robustness and scalability. Django provides a comprehensive set of tools, libraries, and built-in features that make it easy to build complex web applications quickly. It follows the "batteries included" philosophy, offering functionalities such as an ORM (Object-Relational Mapping), authentication, and admin interface out of the box. Django's emphasis on convention over configuration and its strong community support contribute to its ease of use and development speed.
Disadvantage 1:	React: It can be less suitable for small or static websites. React is optimized for building large-scale, complex applications with dynamic user interfaces. If you're working	Angular: it has a steep learning curve and complexity. Angular has a relatively large and complex codebase compared to other frontend frameworks. It introduces

	on a simple website with mostly static content and minimal interactivity, using React may introduce unnecessary complexity and overhead. React's virtual DOM and component-based architecture, while powerful for managing complex UIs, can add extra complexity and performance overhead for simple websites. In such cases, using a simpler framework or even plain HTML, CSS, and JavaScript might be more appropriate and efficient.	its own set of concepts and terminologies, such as modules, decorators, and services, which can be overwhelming for beginners. Additionally, the initial setup and configuration of an Angular project can be more involved compared to other frameworks.
Disadvantage 2:	Node.js: it has a single-threaded nature, which can limit its performance in certain scenarios. Since Node.js runs on a single thread, it may need help to efficiently handle CPU-intensive tasks. If your application requires heavy computation or processing, Node.js may not be the best choice. However, Node.js mitigates this limitation by utilizing asynchronous and non-blocking I/O operations, allowing it to handle a large number of concurrent requests efficiently.	Django: it has a steep learning curve, especially for developers who are new to Python or web development frameworks. Django has its own set of conventions, terminology, and concepts that need to be grasped, which can be overwhelming for beginners. Additionally, while Django is highly flexible, it may involve more initial setup and configuration compared to lighter frameworks. This can be perceived as a disadvantage for simple or small-scale projects where a more lightweight framework may suffice.

3. References:

● Citation of all materials that are not originally written by you, including teaching materials in and out of our course.

● You must use the IEEE style properly:

https://www.ieee.org/content/dam/ieeeorg/ieee/web/org/conferences/style_references_manual.pdf

[1] "MongoDB Tutorial," MongoDB tutorial, <https://www.w3schools.com/mongodb/index.php> (accessed Dec. 15, 2023).

[2] "JavaScript Tutorial," JavaScript Tutorial, <https://www.w3schools.com/js/default.asp> (accessed Dec. 15, 2023).

[3] C. Tsang, "Lab7," in *CSC/2720*, Dec. 15, 2023

[4] C. Tsang, "Lab8," in *CSC/2720*, Dec. 15, 2023

[5] C. Tsang, "Lecture 17: MongoDB," in *CSC/2720*, Dec. 15, 2023

4. Appendix

● A workload distribution. We allow more than one person to do a job together, but each member should have a unique workload.

Name & SID	Job Distribution
Ho Ka Chun 1155176593	User Action: <ul style="list-style-type: none">• Show events whose price under a specific number• Show user name in top-right corner and the log-out function• New Feature, which allows the page to switch in light mode and dark mode
Lee Chi To 1155143279	Admin actions: <ul style="list-style-type: none">• CRUD stored event details in the local database• Log out as admin
Lam Hoi Chun 1155192755	User Actions: <ul style="list-style-type: none">(1) Location Table with sorting with number of events(2) Google Map API(3) Location Search
Lau Ho Man 1155157519	<ul style="list-style-type: none">• Admin actions:<ul style="list-style-type: none">◦ CRUD user data (username and password only) in the local database.• Non-user actions:<ul style="list-style-type: none">◦ Log in as user with username and password.◦ Log in as admin using username and password.• Integration
Lam Cheuk Wing 1155158991	User Actions: <ul style="list-style-type: none">4. A separate view for one single location5. Add location to a list of user's favourite locations and see the list in another view. Location id setting