Hong Kong Cultural Activities Web Application

ESTR2106: Building Web Applications
Group Project Report

ESTR2106 Group 2

CHEN Jinheng	1155233369
CHEN Xinyan	1155233438
LIU Xinran	1155233368
WENG Xiangxiang	1155211173
PAN Huangyu	1155211012

December 18, 2025

Abstract

This project presents a web application for viewing and managing cultural events in Hong Kong. The system integrates data from the Leisure and Cultural Services Department (LCSD) [1], providing users with an intuitive interface to browse venues, explore events, manage favorites, and interact with cultural activity information. The application features a React-based frontend [2] with a RESTful Express.js backend [3], utilizing MongoDB [4] for data persistence. Key functionalities include user authentication, location and event browsing with search and filtering capabilities, interactive map visualization, favorite management, and comprehensive admin tools for database management. Repository link: `https://github.com/xx-Weng/ESTR2106_Group2`

Contents

1 Methodology

1.1 List Files

The project follows a monorepo structure, separating the React frontend and Express.js backend.

```
ESTR2106_Group2/
├── client/ ..... React frontend application
│   ├── public/ ..... Static assets
│   ├── src/
│   │   ├── components/ ..... Reusable React components
│   │   ├── context/ ..... React context providers (Auth, Theme)
│   │   ├── pages/ ..... Page components
│   │   │   ├── Admin/ ..... Admin management pages
│   │   │   ├── Auth/ ..... Authentication pages
│   │   │   └── ... ..... Other pages
│   │   ├── assets/ ..... Images and styles
│   │   └── utils/ ..... Utility functions
│   └── package.json
├── server/ ..... Express backend application
│   ├── config/ ..... Configuration files
│   ├── controllers/ ..... Route controllers
│   ├── middleware/ ..... Custom middleware (auth, upload)
│   ├── models/ ..... MongoDB models
│   ├── routes/ ..... API routes
│   ├── services/ ..... Business logic services
│   │   ├── lcsdService.js ..... LCSD data synchronization
│   │   ├── googlePlacesService.js ..... Google Places integration
│   │   ├── ipGeolocationService.js ..... IP geolocation
│   │   └── ...
│   ├── cache/ ..... Cached XML data
│   ├── uploads/ ..... User-uploaded files (avatars)
│   └── package.json
└── package.json ..... Root package.json with monorepo scripts
```

1.1.1 Frontend Files (client/)

- **Core Application:** `src/index.js` (entry point); `src/App.js` (routing and context).
- **Pages:** `HomePage.js`, `LocationList.js`, `EventList.js`, etc.

1.1.2 Backend Files (server/)

- **Application:** `app.js` (Express configuration).
- **Models & Controllers:** `User.js`, `Location.js`, `Event.js`, etc.

1.2 Data Pre-processing

The application processes XML data from LCSD data.gov.hk API [1]. The pipeline includes multi-source fetching with local cache fallback, handling field variations and whitespace trimming, and automating `location.events` arrays using MongoDB aggregation.
Here we mainly focus on two data files:

- "Programme information" from `https://www.lcsd.gov.hk/datagovhk/event/events.xml`, saved as /server/cache/lcsd/programme-latest.xml;
- "Venues of programmes" from `https://www.lcsd.gov.hk/datagovhk/event/venues.xml`, saved as /server/cache/lcsd/venue-latest.xml.

We noticed that different rooms in the same venue were counted as different locations in the source file, but they actually overlapped completely on the map. Therefore, when displaying the location list on the front end, we manually selected 10 venues with far more than 3 events and merged all their rooms. Note that frontend processing does not affect the normal updates of the backend database.

Figure 1: Location Data Structure in Backend

1.3 Required Actions

1.3.1 User Authentication (sign up, sign in and log out)

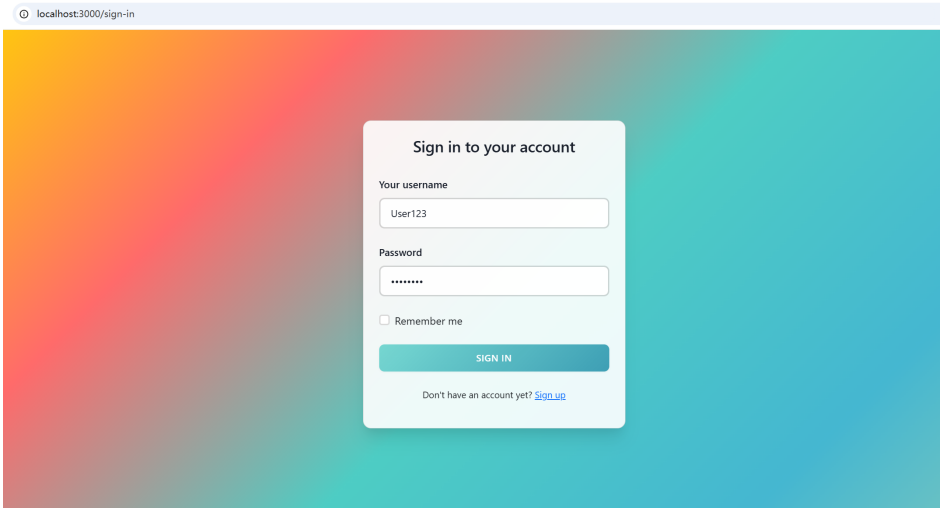

Figure 2: Sign in Interface

Implemented in authController.js and auth.js middleware. JWT tokens stored in HTTP-only cookies provide secure session management. You can find the way to log out by clicking your avatar.

1.3.2 Location Browsing

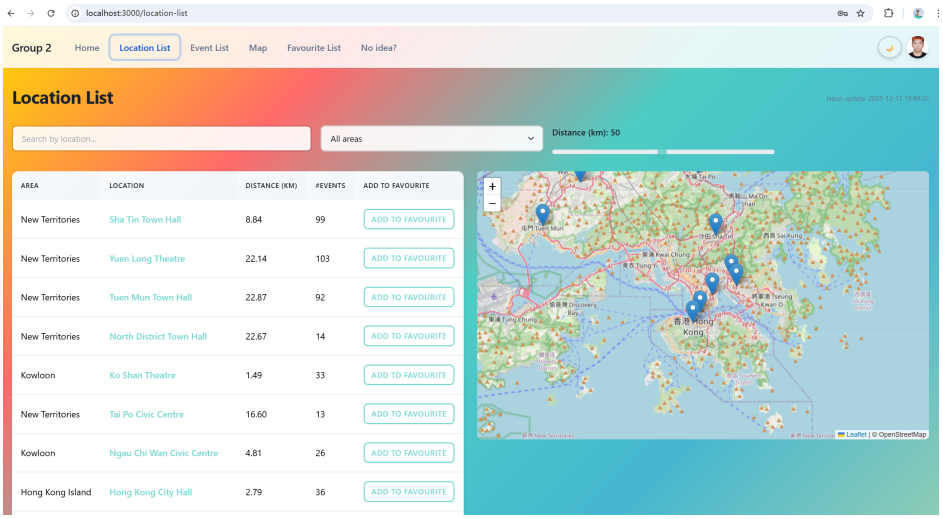

Figure 3: Location List

Implemented in LocationList.js and locationController.js. Displays list of 10 fixed cultural venues with search, filter, and sort capabilities. Accessible from navigation menu "Location List".

1.3.3 Location Details

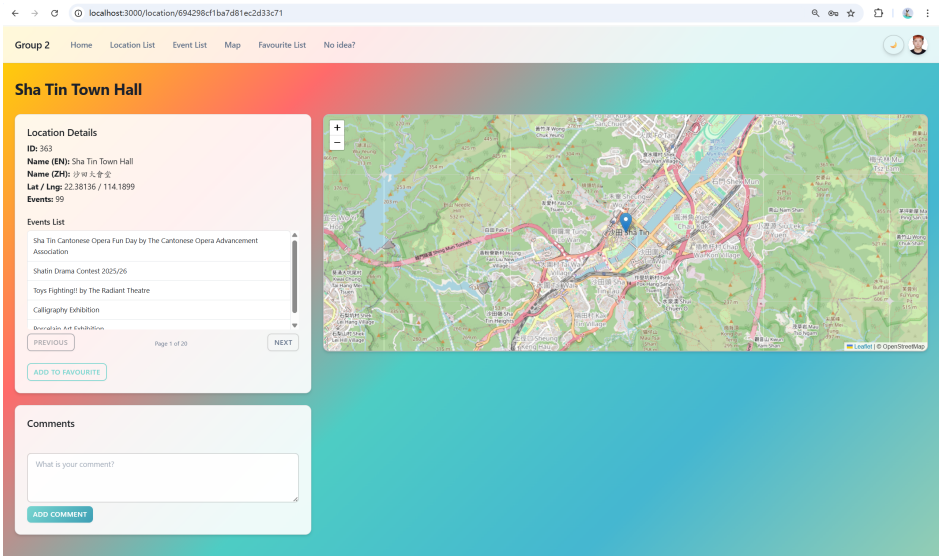

Figure 4: Location Details

Implemented in SingleLocation.js. Displays complete location information, associated events, coordinates, and allows adding/removing favorites and comments. Accessed by clicking locations from the list.

1.3.4 Event Browsing

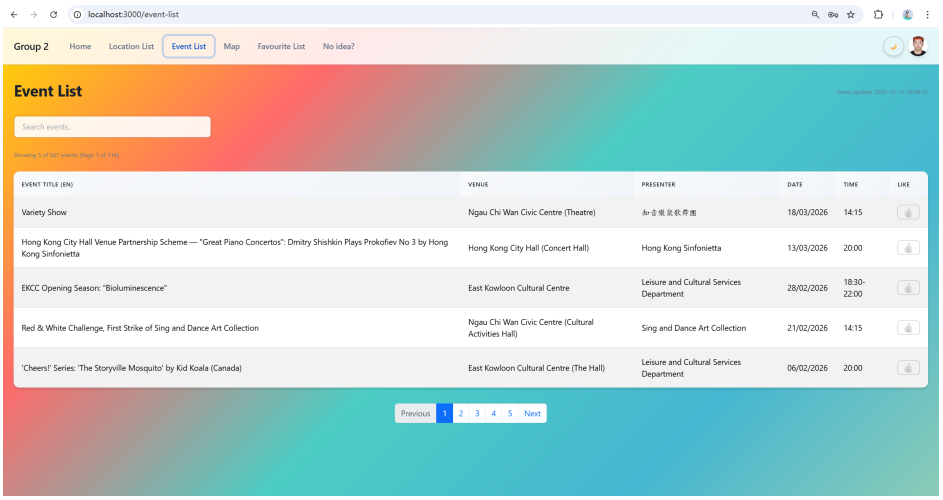

Figure 5: Event List

Implemented in EventList.js and eventController.js. Displays paginated list of all cultural events with search and filtering. Users can view event details and like/unlike events. Accessible from navigation menu "Event List".

1.3.5 Interactive Map

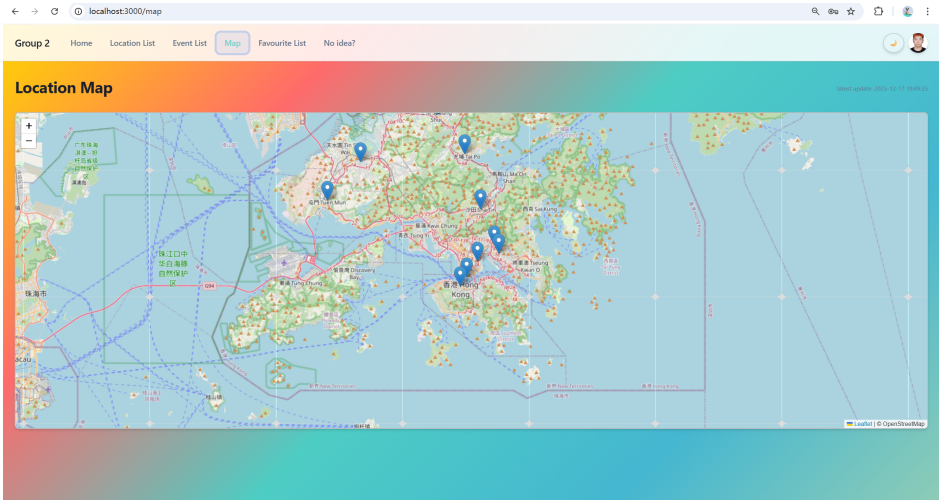

Figure 6: Map

Implemented in MapPage.js using Leaflet [5] library. Displays all venues on interactive map with markers and popup information. Integrates with Google Places API [6] or Nominatim API [7] for additional information. Accessible from navigation menu "Map".

1.3.6 Favorite Management

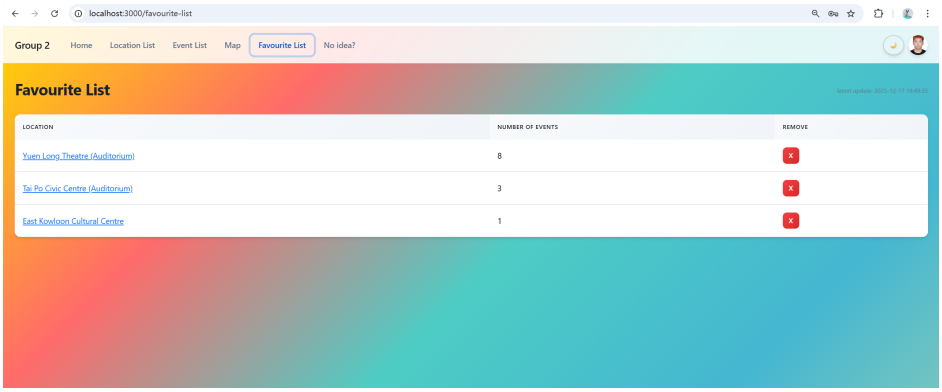

Figure 7: Favourite List

Implemented in FavouriteList.js and locationController.js. Users can add/remove locations to favorites. Favorite status persists across sessions. Accessible from navigation menu "Favourite List".

1.3.7 Admin Panel - User Management

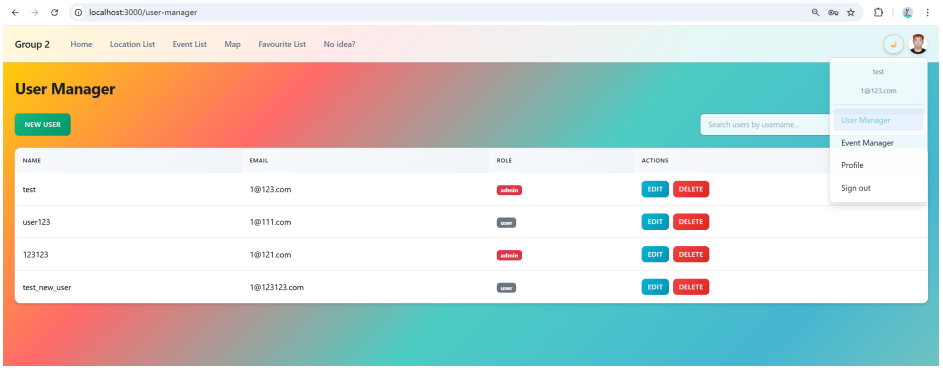

Figure 8: User Manager

Implemented in UserManager.js and userController.js. Admin users can view, create, edit, and delete users. Includes search functionality. Accessible from navigation menu "User Manager" for admin users (click avatar on the top-right corner).

One superadmin account is provided:
Username: superadmin
Password: gpa400

1.3.8 Admin Panel - Event Management

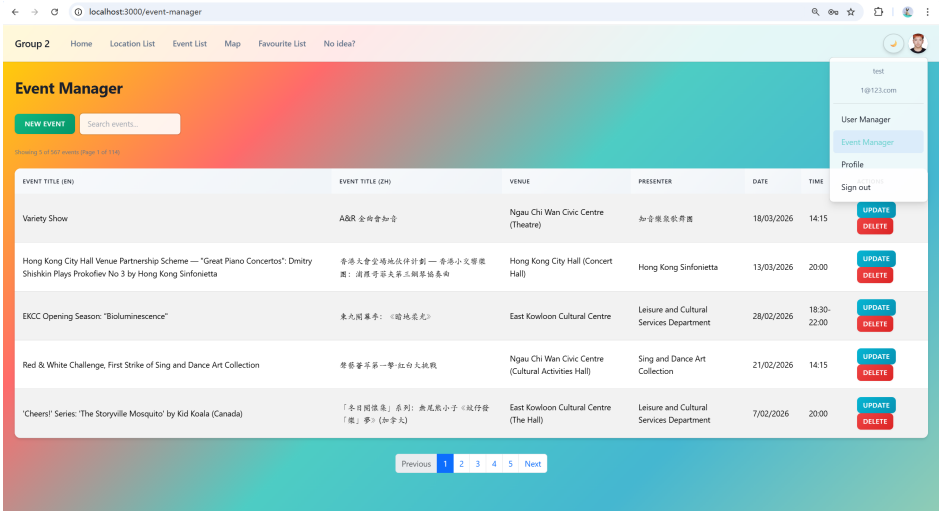

Figure 9: Event Manager

Implemented in EventManager.js and eventController.js. Admin users can view, create, edit, and delete events with pagination and search. Accessible from navigation menu "Event Manager" for admin users (click avatar on the top-right corner).

1.4 Extra Features

1.4.1 Theme Switching

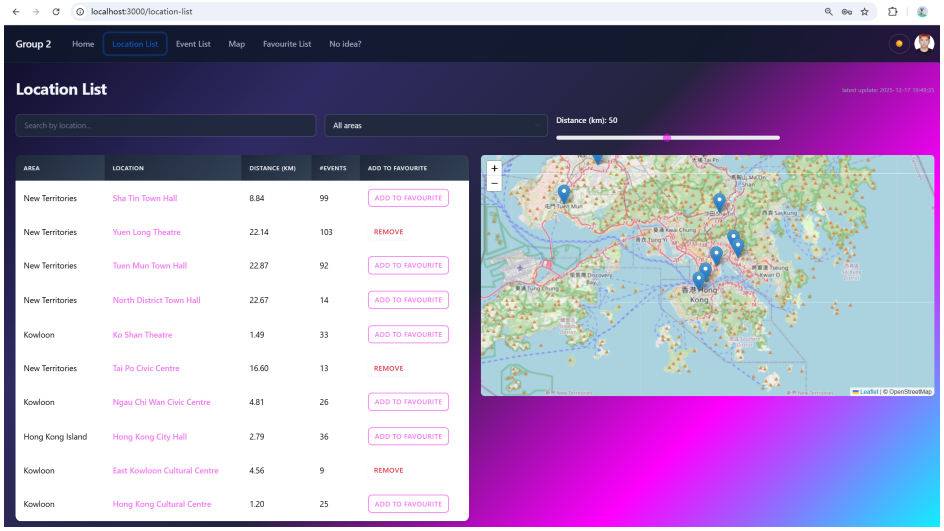

Figure 10: Dark Mode

We maintain a button to switch between light and dark mode. It's near your avatar.

1.4.2 Event Liking System

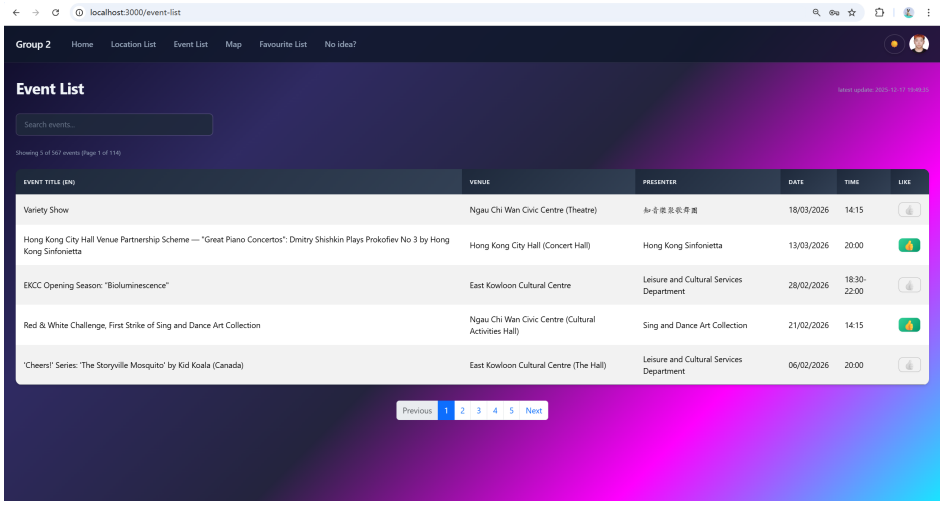

Figure 11: Event Liking System

1.4.3 Responsive Design

Fully compatible with mobile and desktop views.

1.4.4 User Profiles

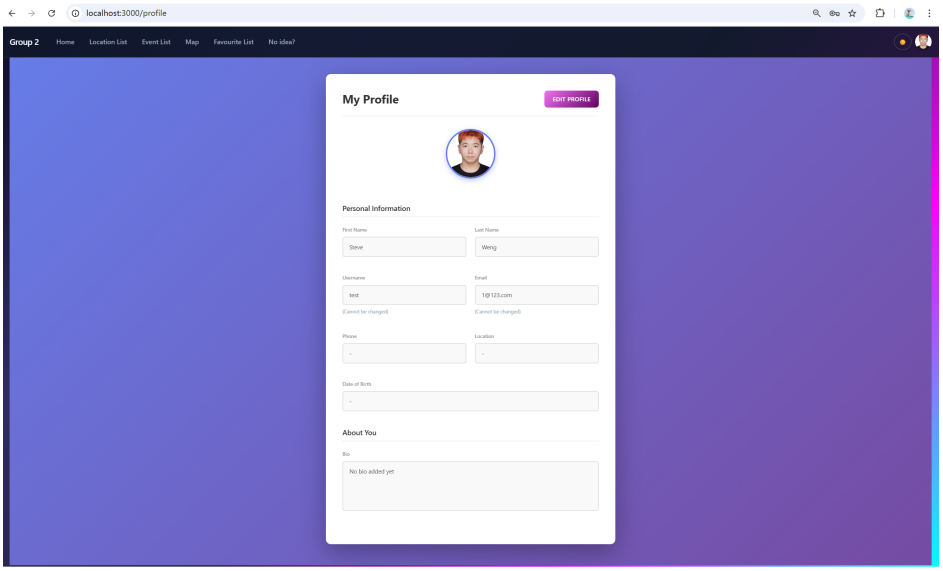

Figure 12: Profile Interface

Users can view and edit their profile information. Supports avatar upload. Implemented in Profile.js. Note: If you have some comments on some location, it will show your avatar. Other users can click into your avatar to see your profile.

1.4.5 Avatar Upload

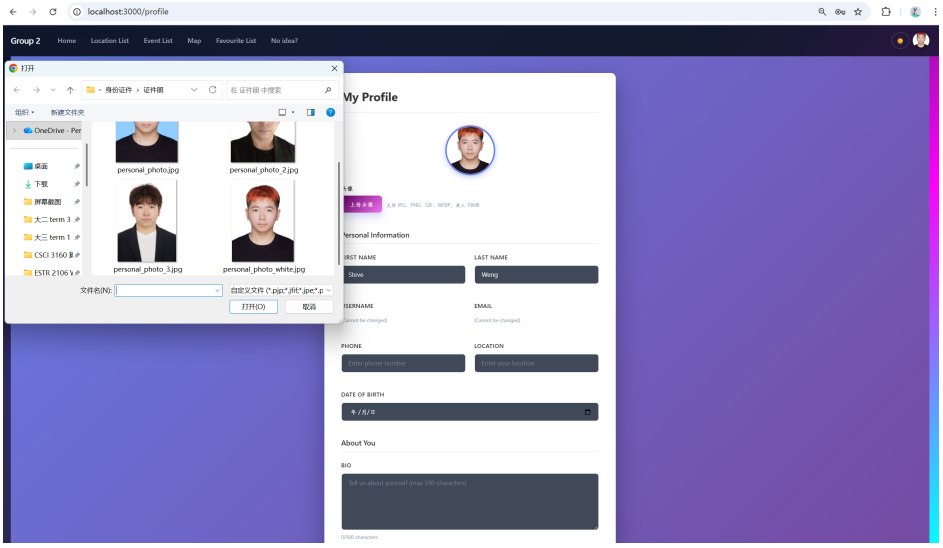

Figure 13: Avatar Upload

Implemented in upload.js middleware using Sharp library.

1.4.6 Other Extra Features

We implemented form collection and form-based activity recommendations on the Homepage. Due to report space limitations, you may try it yourself.

2 Contact Person

Weng Xiangxiang 1155211173@link.cuhk.edu.hk

3 Appendix

CHEN Jinheng, CHEN Xinyan, and LIU Xinran are responsible for front-end development, primarily focusing on the various page files located under /client/src/pages. Weng Xiangxiang and Pan Huangyu are in charge of back-end development, which mainly includes the development of data processing and database connection modules within the /server directory.

We use Git for project collaboration and GitHub for hosting.
Repository link: `https://github.com/xx-Weng/ESTR2106_Group2`

References

[1] LCSD, "Cultural Events," Hong Kong Open Data. [Online]. Available: `https://data.gov.hk/en-data/dataset/hk-lcsd-event-event-cultural`. [Accessed: 18-Dec-2025].

[2] Meta Platforms, Inc., "React Documentation," React. [Online]. Available: `https://react.dev/`. [Accessed: 18-Dec-2025].

[3] Express.js, "Express.js Documentation," Expressjs.com. [Online]. Available: `https://expressjs.com/`. [Accessed: 18-Dec-2025].

[4] MongoDB, Inc., "MongoDB Documentation," MongoDB.com. [Online]. Available: `https://www.mongodb.com/docs/`. [Accessed: 18-Dec-2025].

[5] V. Agafonkin, "Leaflet JS," Leafletjs.com. [Online]. Available: `https://leafletjs.com/`. [Accessed: 18-Dec-2025].

[6] Google, "Google Places API Documentation," Google Developers. [Online]. Available: `https://developers.google.com/maps/documentation/places/web-service/overview`. [Accessed: 18-Dec-2025].

[7] OpenStreetMap Team, "Nominatim API Documentation," Nominatim.org. [Online]. Available: `https://nominatim.org/`. [Accessed: 18-Dec-2025].