COMP3322B Modern Technologies on World Wide Web

Project Two - Event Management System

Total 16 points

Overview:

In this project, you are going to design and implement an event management system to post and manage public and private events. In this project, you will implement the website using different techniques including HTML, CSS, JavaScript, NodeJS, Express, Pug, ReactJS (Optional) and MongoDB. You are supposed to create two database tables for event data and user data. You will be given the sample design of the database tables, and you can design your own database schema if you are interested.

Objectives:

1. A learning activity to support ILO 2.
2. The goals of this programming project are:
   - to get solid experience in using client-side technologies, server-side technologies such as NodeJS and Express, and database system such as MongoDB to design and implement a web-based event management system.

Specifications:

Implement the functionalities of the system, which mainly contains the following parts: 5 pages (the main page showing events, the login page, the register page, the event detail page, the event publishing page), the classification of public events and private events, the differentiated display of past events and current events, the CRUD(Create, Retrieve, Update, Delete) function of events and the join function for public event.

Before developing the website, you should prepare two MongoDB database tables. The suggested database schemas are as follows. You are assumed to have the information for at least 4 users and 10 events in the MongoDB database. Please visit Appendix for the information about the users and events provided by us.

Database 1: Event

Key	Type	Example
eventid	String	"eventid1"
type	String	"public"
title	String	"CS Public Workshop"
starttime	String	"2020-05-01 13:00"
endtime	String	"2020-05-01 18:00"
location	String	"CYC 101"
attenders	[String]	["userid1", "userid2", "userid3"]
description	String	"Workshop description~"
creater	String	"AliceC"
createrid	String	"userid1"

The eventid is the identification of this event. The type can be "public" or "private". The start time field and end time field should contain both the date and time, and the format of time can be decided by yourself. The attenders field is an array of userids, and it contains the

creater by default. The creater field hold the alias name while the createrid field hold the userid of the creater.

Database 2: User

Key	Type	Example
userid	String	"userid1"
name	String	"Alice Cheng"
acctname	String	"AliceC"
email	String	"alice-random@hku.hk"
password	String	"pwd"
sessionid	String	"SYGb0lvjXRHToNX8zFXacw=="
sessiontoken	String	"mA2iY/uOGLh0ti2dTEzYct5/SfbDIlNlGiKz6SPAu3U="

The userid is the identification of the user. The name field contains the real name of the user and the acctname field holds the alias name. The email is used to login the system, so it needs to be exclusive. The sessionid and sessiontoken is used to verify the user and should be generated everytime the user logs in and be cleared while they log out.

Section 1: Create 5 pages (4 points)

1. Main Page (1 point)

Route	GET /main

The page name showed in here is for your reference; you can use a different name, e.g. /main.html

(0.5 points) You are expected to build a main page which shows the public events while the user has not logged in (shown as Figure 1).

* [Title] A Label shows the name of the system "3322 Event Management System". You can make up your own title if you want.
* [Register Button] A button which directs to the register page which will be introduced later.
* [Sign in Button] A button which directs to the login page which will be introduced later.
* [Event List] A table shows the public events while not log in.
    * Title: the title of the event
    * Time: the start time of the event
    * Venue: the location of the event
    * Type: the type of the event ("public" or "private")
    * Owner: the alias name of the creater
    * Attenders: the number of the attenders who are going to join the event.

Figure 1: Sample of the main page when user does not log in

(0.5 points) You are expected to change the main page which shows both the public events and private events of current user <u>when the user logged in</u> (shown as figure 2).

* Register Button is hidden when the user has logged in.
* [Sign out Button] A button for signing out the current user and will direct to the main page without user-specific information (shown as Figure 1).
* [Current Event Button] A button filters the events which will be described later.
* [Past Event Button] A button filters the events which will be described later.
* [Add Event Button] A button directs to the event publishing page which will be introduced later.

Figure 2: Sample of the main page after AliceC logged in the system

2. Login Page (0.5 points)

Route	GET /login

In the Login Page, the items and functions are expected to be as follows:

* [Login] form - A login form which uses the "POST" method to send the data that is entered by the user in the email and password textboxes. Please read section 2.1.
* [Email] textbox - A textbox which allows user to enter the registered email account.
* [Password] textbox - A textbox which allows user to enter the password. The password should not be visible to others when the user is entering the password.
* [Submit] button - A submit button for sending the form data to the server once it is clicked.
* [Register] button - A button which directs to the register page when clicked.

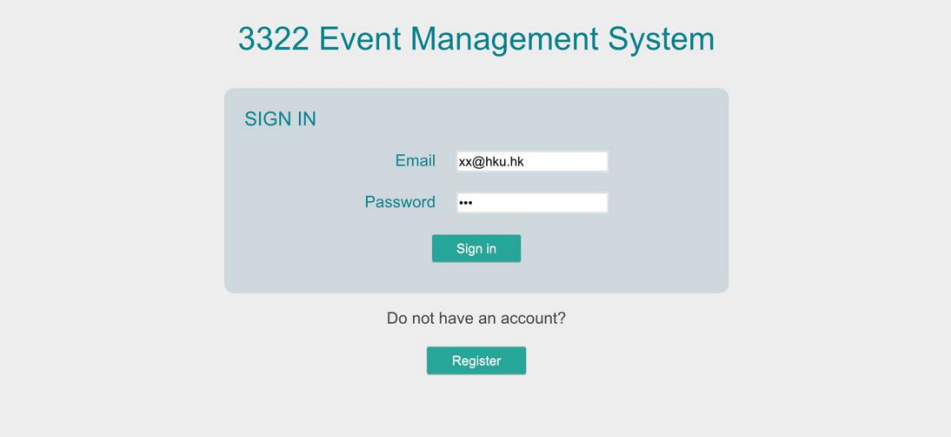

Figure 3: Sample of the login page

3. Register Page (0.5 points)

Route	GET /register

In the Register page, the items and functions are expected to be as follows.

* [Register] form - A login form which uses the method of "POST" to send the data that is entered by the user in the textboxes. Please read section 2.2.
* [Name] textbox - A textbox which allows user to enter the full name.
* [Alias] textbox - A textbox which allows user to enter the nickname.
* [Email] textbox - A textbox which allows user to enter the email account, which is the identification of a user.
* [Password] textbox - A textbox which allows user to enter the password. The password should not be visible to users when the user is entering the password.
* [Confirmation] textbox - A textbox which allows user to reenter the password. It is not visible to users as well.
* [Register] button - A submit button for sending the form data to the server once it is clicked.

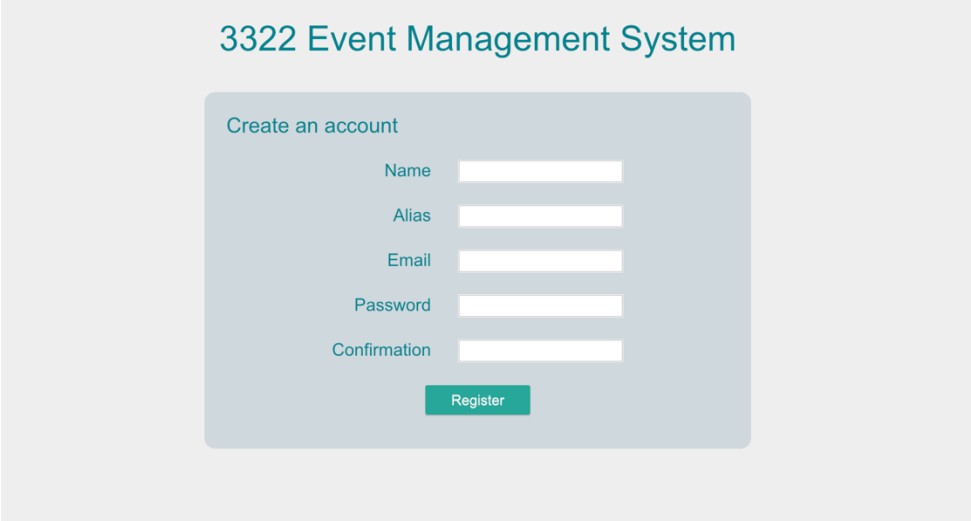

Figure 4: Sample of the Register page

4. Event Detail Page (1.5 points)

Route	GET /events/:eventid

In the event detail page, the items and functions are expected to be as follows.
(0.5 points) when not logged in the system:

* [Back Button] - A button directs to the main page.
* [Register Button] - A button directs to the register page as described before.
* [Sign in Button] - A button directs to the login page as described before.
* [Event Information Table] - A table shows the event detail information include title, type, start time, end time, location, description and the alias name of creater.

Figure 5: Sample of the Event Detail page when user does not log in

(1 point) when the user has logged in the system:
If this is a public event that is not created by the user, a new button will be shown as follows.

* [Join/Leave Button] - A button to join or leave this public event. If he has joined this event, the button shows "Leave" (shown as Figure 6a); If he has not joined this event, the button shows "Join" (shown as Figure 6b).

Figure 6a: Sample of the Event Detail page when user choose a public event that he has not joined

Figure 6b: Sample of the Event Detail page when user choose a public event that he has joined

If the user chooses an event that he/she created, the new buttons will be shown as follows.

* [Delete Button] - A button to delete this event and direct to the main page.
* [Edit Buttons] - Buttons shown next to every item except the creater line.
* [Update] - A button for submitting the new details of the current event.

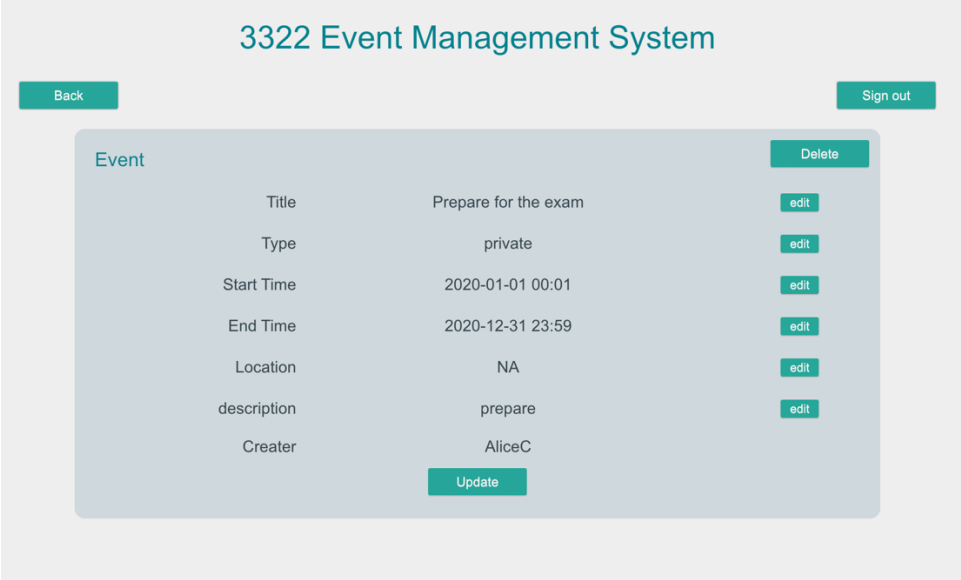

Figure 7: Sample of the Event Detail page when user log in and has chosen the event he created.

5. Event Publishing Page (0.5 points)

Route	GET /addevent

In the Event Publishing page, the items and functions are expected to be as follows.

* [Title] textbox - A textbox which allows user to enter the title of the event.
* [Type] Radio Input - A radio input which allows user to choose whether the event is a 'public' or 'private' event.
* [Start Time] textbox - A textbox which allows user to enter the start time of the event.
* [End Time] textbox - A textbox which allows user to enter the end time.
* [Location] textbox - A textbox which allows user to enter the location information.
* [Description] textbox or textarea - A textbox or textarea which allows user to enter the description information. Limit it to accept at most 200 characters.
* [Submit] button - A submit button for sending the form data to the server once it is clicked.

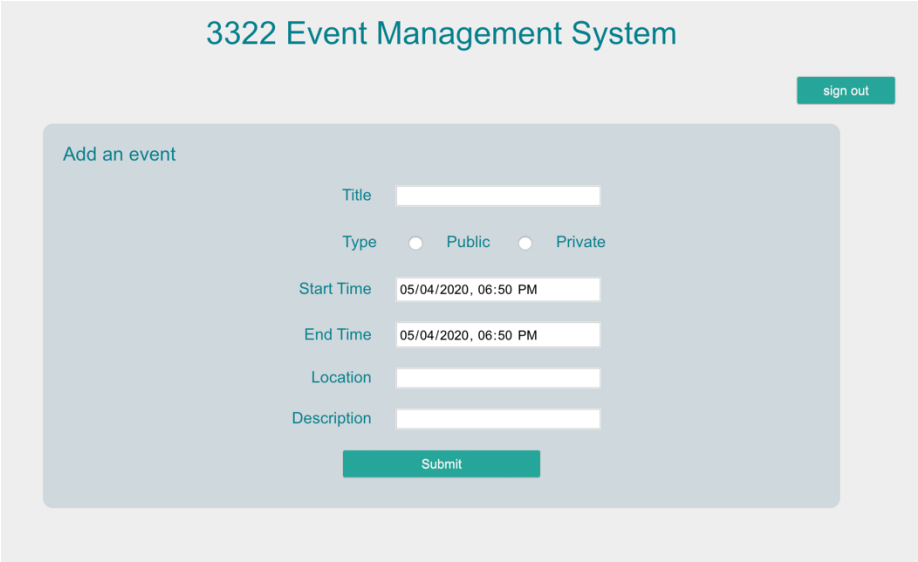

Figure 8: Sample of the event publishing page
(you only need to implement the basic skeleton as stated in the previous page in the current stage)

Section 2: Funtions (7.5 points)

1. Login (0.5 points)

Route	POST /users/signin

In the server side, the functions are expected as follows.

* Check whether the email is in the database, if not, response "User is not registered".
* Check whether the password is matched, if not, response "Unauthorized access".
* If the user logs in successfully, create a new session id and session token which will be used as verification of user-specific operations.

In the client side, the functions are expected as follows.

* If there is an error, alert the message and clear the password.
* If user log in successfully, direct to the main page with this user's specific information (Figure 2).

2. Register (1 point)

Route	POST /users/register

In the server side, the functions are expected as follows.

* Check whether the email is in the database, if so, response "Duplicated user's email address".
* Create a new account with information sent from user and create a userid for this user which will be used as the unique identification of this user. It is also required for creating the new sessionid and session token.

In the client side, the functions are expected as follows.

* Submit the information to the server while the "register" button is clicked.
* If user register successfully, direct to the main page with this user's specific information (Figure 2).

(0.3 points for the below 3 functions)

* Check whether every textbox is filled.
* Check whether the confirmation password is the same as password.
* Check whether the input email conforms to the right format "xx@xx.xx".

3. Main Page (2.5 points)

(1.5 points) Basic presentation of events.

Route	GET /events

In the server side, the functions are expected as follows.

* While no user has logged in, return public events in the database.
* While the user has logged in, return public events and events created by this user in the database.

In the client side, the functions are expected as follows.

* Present the events returned by the server.
* While click on an event in the table, direct to the event detail page showing this event (Section 1.4).

(1 point) Further functions.

The functions are expected as follows.

* Show the events of which the end time is later than now by default.
* When the "Past Event" button is clicked, show the events of which the end time is earlier than now but no earlier than two weeks ago (shown as Figure 9).
* When the "Current Event" button is clicked, show the events of which the end time is later than now by default.
* Sort the events by the starttime. While presenting the past events, events with later start time are shown first. While presenting the current and future events, events with earlier start time are shown on the top.

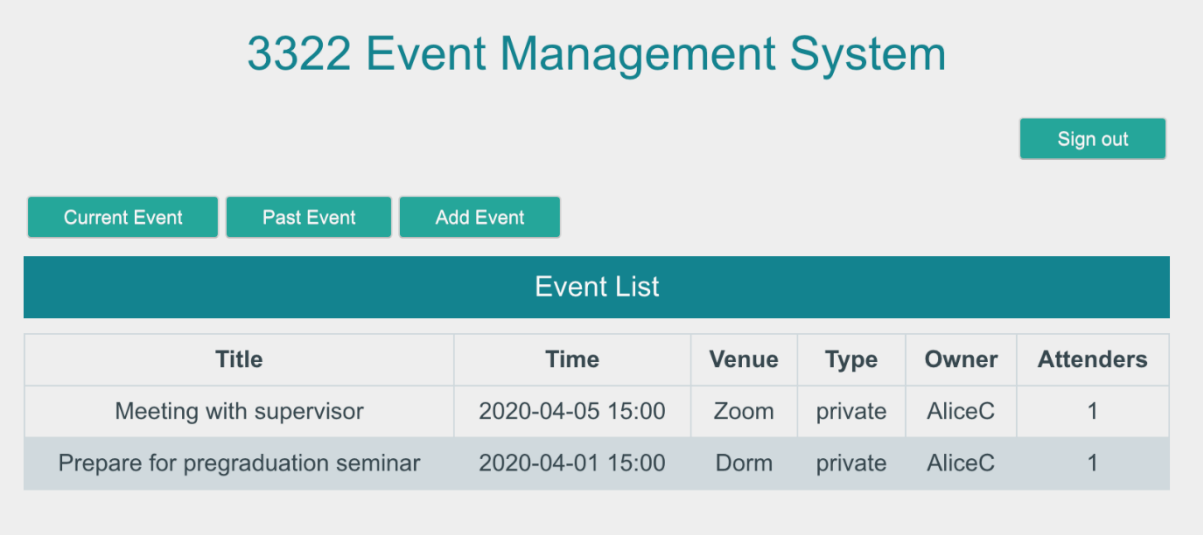

Figure 9: Sample of the past events.

4. Event publishing page (1 point)

(0.5 points) Basic functions

Route	POST /events

In the server side, the functions are expected as follows.

* Add a new event with information sent by users and create an event id as the identification of this event. The attenders should include the creater by default.

In the client side, the functions are expected as follows.

* Check whether every textbox is filled.
* If the new event published successfully, direct to the main page.

(0.5 points) Further functions

The functions are expected as follows.

* Check whether the start time and end time are valid (end time should be later than start time).
* Show the current time in the start time and end time textboxes.

5. Event detail page (2.5 points)

(0.5 points) Delete

Route	DELETE /events/:eventid

In the server side, the functions are expected as follows.

* Delete the event with the event id sent by user.

In the user side, the functions are expected as follows.

* If the event is deleted successfully, direct to the main page in which the deleted event will not be in the list.

(1 point) Edit

Route	PUT /events/:eventid

In the server side, the functions are expected as follows.

* Update the event with the new details sent by the user

In the client side, the functions are expected as follows.

* Click the "Edit" button, there will be a textbox where the user can enter the new details.
* Click the "Update" button, the new details will be sent to the server.
* If the event is updated successfully, direct to the main page.
* Show the original information as the default content in the textbox when the "edit" button is clicked.
* Check the validation of input before submitting the new details.

(1 point) Join/Leave

Route	PUT /events/:eventid/register

In the server side, the functions are expected as follows.

* If the user joins the event, add this user's userid into the attenders field of this event.
* If the user leaves the event, remove this user's userid from the attenders field of this event.

In the client side, the functions are expected as follows.

* If the user joins the event successfully, the button should appear as "Leave".
* If the user leaves the event successfully, the button should appear as "Join".

Section 3: Validation of user operations (2 points)

1. Every user-specific operation should be verified with the session id and session token. The involved operations are as follows.

* Fetch the event list which contains the private events.
* Create, update, delete the event.

2. Every time the user logs in, the server should return the new session id and session token, and if the user signs out, the session id and session token should be cleared in the server.

Section 4: (2.5 points):

(1 point) Apply CSS styling rules to decorate the pages and (1.5 points) implement the responsive web design to make the pages be responsive to different screen dimensions. You are not required to have fancy design because no bonus marks will be given. Thus, just use very standard styling should be fine.

Submission:

The deadline for Project 2 is on May 14, 2020 (Thursday).
You are required to:

1. Submit your work to the course Moodle submission page.
2. To do that, you have to create a project folder "Project2" and place all the program code and the database 'data' folder in this project folder. Use a compression or archive tool to compress the Project2 folder and name the file as Project2-yourStudentNo.zip (or tgz or other common compression formats). Before creating the project zip file, make sure all source files and data for the MongoDB are all placed inside the project folder. [Optional] You can remove the node_modules folder(s) to save space and compression time.
3. Upload a text file to:
    a. List the version numbers of the Node.js, Express.js, and MongoDB installed in your platform.
    b. List any special libraries you have included in your program.
    c. Indicate how to (1) start the MongoDB database, (2) start the web server, and (3) access the main page of the Web application.
    d. Indicate how much work you have completed for the project. You are recommended to make use of the table shown in the Grading Policy section as the checklist. Using it to show us how much you have accomplished.

    Please submit this text file to the course Moodle submission page too.

Grading Policy

Total: 16 points	<u>Create pages (4 points)</u> * Correctness of the main page (1 point) * Correctness of the login page (0.5 points) * Correctness of the register page (0.5 points) * Correctness of the event detail page (1.5 points) * Correctness of the event publishing page (0.5 points) <u>Functions (7.5 points)</u> * Correctness of functions in login page (0.5 points) * Correctness of functions in register page (1 point) * Correctness of functions in main page (2.5 points) * Correctness of functions in event publishing page (1 point) * Correctness of functions in event detail page (2.5 points)

	Validation of user operations (2 points) Styling and responsive design (2.5 points)

Plagiarism

Plagiarism is a very serious academic offence. Students should understand what constitutes plagiarism, the consequences of committing an offence of plagiarism, and how to avoid it. **Please note that we may request you to explain to us how your program is functioning as well as we may also make use of software tools to detect software plagiarism.**

Appendix A: User Infomation

User 1:

Key	Value
userid	"userid1"
name	"Alice Cheng"
acctname	"AliceC"
email	"user1 @hku.hk"
password	"pwd"
sessionid	"s-id1"
sessiontoken	"s-token1"

User 2:

Key	Value
userid	"userid2"
name	"Emma Wong"
acctname	"E.W."
email	"user2 @hku.hk"
password	"pwd"
sessionid	"s-id2"
sessiontoken	"s-token2"

User 3:

Key	Value
userid	"userid3"
name	"John Lee"
acctname	"John"
email	"user3 @hku.hk"
password	"pwd"
sessionid	"s-id3"
sessiontoken	"s-token3"

User 4:

Key	Value
userid	"userid4"
name	"Bob Parker"
acctname	"BB"
email	"user4 @hku.hk"
password	"pwd"
sessionid	"s-id4"
sessiontoken	"s-token4"

Appendix B: Event Information

Event 0

Key	Value
eventid	"eventid0"
type	"private"
title	"Prepare for the exam"
starttime	"2020-01-01 00:01"
endtime	"2020-12-31 23:59"
location	"N/A"
attenders	["userid1"]
description	"prepare"
creater	"AliceC"
createrid	"userid1"

Event 1

Key	Value
eventid	"eventid1"
type	"public"
title	"CS Public Workshop"
starttime	"2020-05-01 13:00"
endtime	"2020-05-01 18:00"
location	"CYC 101"
attenders	["userid1", "userid2", "userid3"]
description	"Workshop description~"
creater	"AliceC"
createrid	"userid1"

Event 2

Key	Value
eventid	"eventid2"
type	"public"
title	"Online Job Fair"
starttime	"2020-04-01 8:00"
endtime	"2020-05-01 18:00"

location	"Zoom"
attenders	["userid1", "userid2", "userid3"]
description	"Job Fair for CS students"
creater	"AliceC"
createrid	"userid1"

Event 3

Key	Value
eventid	"eventid3"
type	"public"
title	"CS Workshop 2"
starttime	"2020-04-30 13:00"
endtime	"2020-04-30 18:00"
location	"CYC 101"
attenders	["userid1", "userid3"]
description	"workshop 2"
creater	"AliceC"
createrid	"userid1"

Event 4

Key	Value
eventid	"eventid4"
type	"public"
title	"Group Meeting"
starttime	"2020-04-10 13:00"
endtime	"2020-04-10 15:00"
location	"Zoom"
attenders	["userid1", "userid2"]
description	"Group Meeting"
creater	"AliceC"
createrid	"userid1"

Event 5

Key	Value
eventid	"eventid5"
type	"private"
title	"Meeting with supervisor"
starttime	"2020-04-05 15:00"
endtime	"2020-04-05 18:00"
location	"Zoom"
attenders	["userid1"]
description	"Regular meeting"
creater	"AliceC"
createrid	"userid1"

Event 6

Key	Value
eventid	"eventid6"
type	"private"
title	"Prepare for pregraduation seminar"
starttime	"2020-04-01 15:00"
endtime	"2020-04-05 15:00"
location	"Dorm"
attenders	["userid1"]
description	"Prepare for pre-graduation Seminar"
creater	"AliceC"
createrid	"userid1"

Event 7

Key	Value
eventid	"eventid7"
type	"public"
title	"Pre-graduation Seminar: Emma Wong"
starttime	"2020-04-06 15:00"
endtime	"2020-04-06 16:00"
location	"CYC 101"
attenders	["userid1", "userid2"]
description	"Host: Emma Wong"
creater	"E.W."
createrid	"userid2"

Event 8

Key	Value
eventid	"eventid8"
type	"public"
title	"Online Competition: Warm-up"
starttime	"2020-04-01 00:00"
endtime	"2020-04-10 23:59"
location	"CYC 101"
attenders	["userid", "userid2", "userid3"]
description	"Warm-up"
creater	"John"
createrid	"userid3"

Event 9

Key	Value
eventid	"eventid9"
type	"public"
title	"Online Competition"

starttime	"2020-04-01 00:01"
endtime	"2020-05-01 23:59"
location	"Online: www.codeforces.com"
attenders	["userid2", "userid3"]
description	"Online Coding"
creater	"John"
createrid	"userid3"

Event 10

Key	Value
eventid	"eventid10"
type	"private"
title	"Do homework"
starttime	"2020-04-07 17:00"
endtime	"2020-04-07 18:00"
location	"Main Library"
attenders	["userid4"]
description	"Homework of COMP 0001"
creater	"BB"
createrid	"userid4"

Event 11

Key	Value
eventid	"eventid11"
type	"public"
title	"Bob s birthday party"
starttime	"2020-05-01 18:00"
endtime	"2020-05-01 20:00"
location	"Park Hotel"
attenders	["userid1", "userid2", "userid3", "userid4"]
description	"Bob s birthday"
creater	"BB"
createrid	"userid4"

Event 12

Key	Value
eventid	"eventid12"
type	"public"
title	"Class COMP 0001"
starttime	"2020-05-30 13:00"
endtime	"2020-05-30 15:00"
location	"CYC 001"
attenders	["userid4"]
description	"Attend class"
creater	"BB"

createrid	"userid4"